The International Congress for global Science and Technology

ICGST International Journal on Artificial Intelligence and Machine Learning (AIML)

Volume (15), Issue (I)
December, 2015

www.icgst.com
www.icgst-amc.com
www.icgst-ees.com

AIML Journal
ISSN Print 1687-4846
ISSN Online 1687-4854
ISSN CD-ROM 1687-4862

Table of Contents

ICGST International Journal on Artificial Intelligence and Machine Learning (AIML)

A publication of the International Congress for global Science and Technology - (ICGST)

ICGST Editor in Chief

Dr. rer. nat. Ashraf Aboshosha

www.icgst.com, www.icgst-amc.com, www.icgst-ees.com

editor@icgst.com

www.icgst.com

A novel Hybrid Search for Minimal Perturbation Problems based on Backjumping and Dynamic backtracking methods

EL GRAOUI EL MEHDI[1], BENELALLAM IMADE[2], BOUYAKHF EL HOUSSINE[1]
[1]LIMIARF, Department of Physics, Faculty of Sciences, Mohammed V University, Rabat, Morocco
[2]National Institute of Statistics and Applied Economic, Irfane Rabat, Morocco
elgraouimehdi@gmail.com, imade.benelallam@ieee.org, bouyakhf@mtds.com

Abstract

Many real-life problems in Artificial Intelligence (AI) as well as in other areas can be efficiently modeled and solved using constraint programming techniques. In many real-life scenarios the problem is partially dynamic. For example, once a change appears in the environment, after the original problem resolution, this change should be reflected in the new solution. The minimal perturbation problem considers such changes, as well as the initial solution to define a new problem whose solution should be as close as possible to the initial solution.

In this paper, we propose two new approaches: HS MPP backjumping and HS MPP dynamic backtracking. These algorithms are based on HS MPP approach (Hybrid Search for Minimal Perturbation Problem) [1]. They rely on the intelligent backtracking methods, namely the backjumping and dynamic backtracking which allow reducing the number of constraints tested and thus the computational time. The evaluation of performance is applied for random binary problems and meeting scheduling problems, with the criteria of computation time, number of constraints checks and number of visited nodes. Finally, the empirical results with these search methods show the efficiency of our proposed algorithms.

Keywords: *Intelligent backtracking, back-jumping, HS MPP, HS MPP BJ, dynamic backtracking, HS MPP DB, minimal perturbation problem, Constraint Satisfaction Problem (CSP), Meeting Scheduling Problem (MSP).*

Nomenclature

HS_MPP	Hybrid Search for Minimal Perturbation Problem
BJ	Backjumping
DB	Dynamic Backtracking
CSP	Constraint Satisfaction Problem
MPP	Minimal Perturbation Problems
AI	Artificial Intelligence
CP	Constraint Programming
MAC	Maintains Arc Consistency
NP-hard	Non-deterministic Polynomial-time hard
GAC	Generalized Arc Consistency
PDB	Partial-order Dynamic Backtracking
DCSP	Dynamic Constraint Satisfaction Problem
SVA	Solution Value Assignments
EPDB	Extended Partial-order Dynamic Backtracking
UB	Upper Bound
MSP	Meeting Scheduling Problem (MSP)
CCs	Constraints Checking
Vn	Visited Nodes
Ts	Computation time in s

1. Introduction

Most existing CP applications are designed for static problems. These ones can be expressed, and solved by appropriate techniques; the solution is applied without any change to problem statement. In practice, the problem formulation is not static but it evolves in time. In fact, it evolves even during solving the problem. Thus, to further spread up applicability of the constraint satisfaction technology in real-life applications, it is necessary to cover the dynamics of the real world. In particular, it is necessary to handle changes in the problem specification during the solving process. The problem changes may result from the changes in the environment like course timetabling, delayed flights, and other unexpected events. The users may also initiate some other changes that might specify new properties and specifications of the problem based on a (partial) solution found so far. The main goal is to find an optimal solution for the users. Naturally, the problem

solving process should continue as smoothly as possible after any change in the problem formulation. In particular, the solution of the altered problem should not differ much from the solution found for the original problem. There are several reasons to keep the new solution as close as possible to the existing solution. For example, if the solution has already been published like the assignment of gates to flights then it would not be convenient to change it frequently because it would confuse passengers. Moreover, the changes to the already published solution might force other changes because the originally satisfied wishes of the users may be violated, which raise an avalanche reaction.

The above type of dynamic problems called a minimal perturbation problem (MPP). Its basic task is to find a solution of the altered problem in such a way that this new solution does not differ much from the solution of the original problem.

The minimal perturbation problem is first comprehensively included in the context of general dynamic scheduling problems. Recently a new approach, called HS MPP, has been proposed [1] for finding the most similar solution to a changing constraints satisfaction problem. The proposed method exploits the fact that the goal is to find a complete assignment with two different properties. The first is that it contains as many as possible identical assignments to the previous solution. The second is that the assignment must be consistent, i.e. it must be a satisfying solution to the new CSP. The HS MPP algorithm presents its efficiency when it reaches the optimal solution. Nevertheless, in some problems it requires an enormous computing time to find this solution.

This is due to the thrashing phenomenon of chronological backtracking; the same failure can be rediscovered a great number of times. This makes it impossible to find the solution more quickly. The backtracking of the algorithm HS MPP deletes the last variable of Current Assignment even if is not responsible for the failure. For that purpose, we have proposed to modify the chronological backtracking by intelligent reasoning techniques to delete the variables that prevent the MAC algorithm to find the optimal solution quickly. The advantages of our new algorithms are illustrated in the experiments section.

After introducing the notion of dynamic problems in section one, each subsequent is laid out as follows. Section 2 describes the minimal perturbation problem. Sections 3 and 4 present our approaches. Section 5 treats experimental results. Finally, section 6 summarizes the advantages of our approaches.

2 Related works

DCSPs, as defined in [8], are a series of CSPs that differ one from the other in some of their constraints. Solving a DCSP is researching a solution for each CSP in the series. A number of methods for acquiring a new solution for a changed CSP were proposed, like nogoods, dynamic backtracking and local search. However, few of the proposed methods are guaranteed to find the most similar solution to the previous one (when problems have multiple solutions).

The authors in [2] propose the unimodular probing algorithm based on constraint programming techniques, which leverages the efficiency of linear programming to solve part of the problem. The aim of this approach focuses on reconfiguring schedules in response to a changing environment. In [3] the approach is one of the first studies in minimal perturbation problems in the context of university course timetabling, using a constraint satisfaction heuristic combined with a branch and bound process. In [4] the authors present a course structure model which is implemented at Purdue University, USA. This structure is based on an iterative forward search algorithm, supporting dynamic aspects of the minimal perturbation

In [9], authors propose a method that, given a revised CSP and a solution to the original problem, finds a solution to the new CSP that is the most similar to the previous solution. The authors establish that this method is feasible only when the changed constraints are unary, and that it causes a significant slowdown in comparison with an algorithm that searches for any solution to the new problem. In a later study, they offer approximation feasible methods that do not guarantee the finding of the most similar solution [10].

Different aspects of finding similar and diverse solutions to specific assignments for CSPs, have been studied in [11]. They prove that finding a solution to a CSP that is rather close to (or enough distant from) a given set of assignments is NP-hard (actually they prove it is FPNP ([log n] - complete). They also propose an algorithm, which is based on Branch and Bound and generalized arc consistency (GAC) for finding the closest solution to a given set of assignments. They propose, in [10], an algebra that enables a combination of similarity constraints to a set of ideal partial assignments, as well as distance constraints from non-ideal partial assignments.

In [11], the minimal perturbation problems MPP are studied for classic scheduling problems. The authors propose an integration of mathematical programming techniques with constraint programming in order to speed-up the search on this special optimization problem. The solutions are used within the constraint program. The authors extend this approach in [12] and discuss its potential to detect sub-problems, which can be solved more efficiently in different CSP applications.

In [3], they propose a formal model of the MPP based on the CSP model and a Branch and Bound algorithm for solving it. The originality in the proposed algorithm is that it allows incomplete solutions to the problem.

In [1], authors adopt the formalism proposed in [3], but seek to find complete solutions to the problem with minimal perturbation as in [11].


In other hand, repairing solutions in DCSPs is studied in [13]. Authors propose EPDB for dynamically changed environments. This approach, which is an extension of the Partial-order Dynamic Backtrack (PDB) [14], exploits PDB flexibility to repair solutions by using retroactive data structures: safety conditions and nogoods, saved previously in PDB process, to guide the search process in a dynamic environment according to based-repair heuristic approaches. Authors prove that EPDB allows DCSPs to be dealt efficiently.

3 Backgrounds

3.1 Minimal Perturbation Problem

A Constraint Satisfaction Problem (CSP) is represented by a triple (V, D, C) Where:

- V = V1,V2,…,Vn is a finite set of variables;
- D = D1, D2, …, Dn is a set of domains, where Di is a set of possible values for the variable Vi;
- C=C1, C2,…, Cm is a finite set of constraints restricting the values that the variables can simultaneously take.

To define the minimal perturbation problem (MPP), we consider an initial (original) problem, its solution, a new problem, and some distance function which allows us to compare solutions of the initial and the new problem. Afterwards we look for a solution of the new problem with minimal distance from the former solution. The original and the new problem can be defined as a CSP.

Let I be a complete assignment for C. The key objective of the MPP is to find an assignment A such that all of the constraints are satisfied, and the number of variables in A whose value differs from I is minimized. In [1] the value of variable V in the original assignment is called the Starting Variable Assignment of V, or its SVA. While previous works, defined the MPP more generally, i.e., a general distance function, and a partial initial assignment, the lower bound functions used in the previous work assume the definition above, and the actual experimental evaluation of the previous algorithms were performed on binary CSPs based on this definition. Finally, the MPP is a triplet $\pi=(\Theta,\alpha,\delta)$,

where:

- Θ is a CSP;
- α is a partial assignment which is called initial assignment;
- δ is a function which defines a distance between any two assignments.

A solution to a MPP problem is a solution to π with minimal distance from α according to δ.

3.2 HS MPP description

For relevant purposes, in the following, we will keep as much as possible the same illustrations presented by the original authors in [1]. The hybrid search algorithm for the minimal perturbation problem (HS MPP) includes two phases which are interleaved throughout the search. In the first phase the algorithm assigns SVAs to variables. This generates a partial solution which includes only SVAs, i.e., a partial solution σ to π with $\delta=0$ (zero distance between α and σ). This phase of the algorithm implements a branch and bound scheme where the upper bound is the smallest δ among the solutions to Θ that were found so far by the algorithm. If the algorithm detects that the partial solution, σ cannot be extended to a solution with a δ smaller than the current upper bound, it backtracks. In order to detect the need to backtrack as early as possible, the algorithm uses an admissible heuristic function described in [1]. If no more SVAs can be assigned to unassigned variables (i.e., all SVAs of unassigned variables conflict with assigned SVAs) and the admissible heuristic does not breach the upper bound, the second phase of the algorithm is performed. In the second phase, a Maintaining Arc Consistency algorithm [6] (MAC algorithm) is performed in order to validate that the partial solution σ with $\delta = 0$, found in the first phase, can be extended to a complete solution. If the second phase ends successfully and a solution γ is found, it is recorded as the best solution found so far and the upper bound is set to $\delta(\gamma,\alpha)$. The other option is that the satisfaction algorithm finds that σ cannot be extended to a complete solution to Θ. In both cases, after this phase is completed, the algorithm resumes the first phase by removing the last SVA assignment (i.e. backtracking).

4 Intelligent backtracking algorithms for MPP

Backtracking is a primary method of systematic search for constraint satisfaction problems. It consists of a blind search for the solution by trying successively all the affectations of variables until it finds a solution [7]. In every partial affectation, the algorithm tests the constraints and as soon as a partial affectation is inconsistent, a backtracking is made.

This technique often suffers from a phenomenon "trashing", reflected by the fact of rediscovering in a repetitive way the same failures as well as the same partial assignments during the search.

Methods called "retrospective: intelligent backward reasoning" trying to identify the causes of these failures. They take advantage of this information to select improved return point.

In this section we present two new intelligent backtracking based approaches for minimal perturbation problems, called HS MPP BJ and HS MPP DB.

Figure 3 presents the main part of corrected Hybrid Search MPP algorithm for backjumping and dynamic backtrack


```
Algorithm 1 : HS MPP
1: upper bound ← n
2: t ← false
3: solution ← null
4: current_assignment ← ∅
5: unassigned_variables ← Θ.v
6: while(phase1)
7:    phase2
8: return solution
phase 1
9:   while(assign_SVA)
10:   while(n-(|current_assignment|+heuristic) ≥ upper_bound)
11:       if(|current_assignment |> 0)
12:            backtrack
13:       else
14:            return false
15: if(current_assignment=null and t)
16:    return false
17: else
18:    return true
phase 2
19: if(current_assignment=null)
20:    t ← true
21: current_solution ← MAC(current_assignment)
22: If(current_solution ≠ null)
23:    upper_bound ← δ(current_solution,α)
24:    solution ← current_solution
25: do
26:   backtrack
27: while(n-(|current_assignment|+heuristic) > upper_bound and |current_assignment|>0)
```

Figure1. Main part of the corrected Hybrid Search MPP algorithm.

```
Algorithm 2 : backtrack
1: if(current_assignment=null)
2:   v ←current_assignment.last
3:   remove_SVA(v)
4:   for each(v' ∈ v.removed_conflicting_SVAs)
5:    move_back_to_domain(v'.SVA)
6:    v.removed_conflicting SVAs ←v.removed_conflicting_SVAs \v'
7:   unassigned_variables.add(v)
```

Figure2. Functions used by the corrected hybrid search MPP algorithm (for assign _SVA, remove_SVA, select_SVA_var and heuristic see Fig2 in [1]).

4.1 HS MPP_BJ description

In the original HS MPP algorithm, when assigning a variable is unsuccessful, backtracking performs a simple backtrack into recently instantiated variable. A well-designed conflicts set analysis will allow a jump towards a higher level (backjumping) [15][16], if it turns out that the cause of the failure is upstream in the tree search.
For that reason, we proposed a new approach, called HS_MPP_BJ, which combines the principle of the algorithm of search for the optimal solution HS_MPP as well as the advantages of the backjumping.

In Zivan and al [1], the phase 1 of the algorithm HS MPP begins by choosing the first variable of the current assignment then it deletes the SVA of the variables which are in conflict with this first variable. These instructions are performed for all the variables of the current assignment. The phase 1 ends when all the variables have no more SVA.
This time HS MPP resorts to the phase 2 in order to solve the problem. The MAC algorithm looks for the solution of the problem. If the problem is solved, this solution is registered then the algorithm tries to look for a better solution than the one already found. Otherwise, backtracking removes the last variable in the current assignment and its SVA because this variable is considered as the culprit variable that forbids the current partial SVAs to be extended. Nevertheless, this deleted variable does not represent necessarily the responsible variable for the algorithm failure.

To remedy this problem, we suggested in our approach HS MPP_BJ to modify the chronological backtracking by the backjumping technique to delete the culprit variable which causes the problem as well as the other variables which follow it in the current assignment. When our algorithm finds the first solution by means of the backjumping, it reuses the backtracking to look for the optimal solution.

We illustrate the following example to compare between both algorithms: HS MPP and HS MPP_BJ.

4.2 HS MPP_DB description

The dynamic backtracking is a sophisticated method of the standard backtracking [8]. It tries to memorize the causes of the failures to avoid reproducing them during the exploration of the conditions of these failures.
Previously, we presented our approach HS_MPP_BJ based on the backjumping method.

In a similar way, we are going to propose the second approach HS MPP_DB which uses the dynamic backtracking instead of the backjumping. For reasons of similarity of the algorithms, we are going to introduce only the difference between these two approaches.


```
Algorithm 5 : HS MPP_BJ or HS_MPP_DB
1: upper bound ← n
2: t ←false
3: solution ← null
4: current_assignment ←∅
5: unassigned_variables ←Θ.v
6: while(phase1)
7:   phase2
8: return solution
phase 1
9:  while(assign_SVA)
10:  while(n-(|current_assignment|+heuristic) ≥ upper_bound)
11:     if(|current_assignment |> 0)
12:        backtrack
13:     else
14:        return false
15: if(current_assignment=null and t)
16:   return false
17: else
18:   return true
phase 2
19: if(current_assignment=null)
20:   t ← true
21: current_solution ←MAC(current_assignment)
22: If(current_solution ≠ null)
23:   upper_bound ←δ(current_solution,α)
24:   solution ←current_solution
25: do
26 :  if(solution = null)
27 :     backjumping or dynamic_backtrack
28 :  else
29 :     backtrack
30: while(n-(|current_assignment|+heuristic) > upper_bound and |current_assignment|>0)
```

Figure3. Main part of the corrected Hybrid Search MPP algorithm for backjumping and dynamic backtrack.

At the level of the phase 2, the backjumping of the algorithm HS MPP_BJ deletes the variable which causes the problem as well as the variables which follow it in the current assignment. Indeed, most of the variables got deleted in the phase 2 by the backjumping are put back in the current assignment in the later phase (phase 1) by the algorithm, i.e. the computational time increases due to the number given variables.

In the new proposed approach HS MPP_DB, only the variable that causes the problem is deleted from the current assignment. This reduces the computational time compared to HS MPP_BJ algorithm. We will illustrate a comparative example of three algorithms: HS MPP, HS MPP_BJ and HS MPP_DB.

4.3 Example of program execution

An execution of HS MPP, HS MPP_BJ and HS MPP_DB on a simple minimal perturbation problem (Figure 6) is shown in presents the solution of the original problem with:

- Four variables v = {v0; v1; v2; v3};
- All domains include three values Di = {1; 2; 3; 4};
- And the constraints v0 ≠ v1, v0 > v2, v1 = v2, v1 > v3 and v2 ≠ v3.

```
Algorithm 3 : backjumping
1 : v ← null , j ← 0
2 : for each(v' ∈ current_assignment )
3 :    if(v' = S1)
4 :       v ← v'
5 :       j ← index(v')
6 :    else if(v'=S2)
7 :       v ← v'
8 :       j ← index(v')
9 :    if(v ≠ null )
10 :      remove-SVA(v,j-1,false)
11 :      for each(v' ∈ v.removed-conflicting-SVAs)
12 :        for each(v" ∈ unassigned-variables)
13 :          if(v= v")
14 :            v".valeur_domain.add(v".SVA)
15 :      v.removed_conflictiong_SVAs.clear()
16 :      current_assignment.remove(j)
17 :      for each(v' ∈ cuurent_assignment tq index(v'>j))
18 :        for each(v" ∈ v′.removed_conflicting-SVAs)
19 :          foreach(s ∈ unassigned_variables)
20 :            if(v"=s)
21 :              s.valeur-domain.add(s.SVA)
22 :      v′.removed-conflicting-SVAs.clear()
23 :      unassigned-variables.add(v′);
24 :      current-assignment.remove(v′);
25 :     unassigned-variables.add(v);
26 :     else
27 :       backtrack
```

Figure4. Backjumping function

```
Algorithm 4 : dynamic-backtrack
1 : v  null , j  0
2 : for each(v' ∈ cuurent_assignment)
3 :    if(v' = S1)
4 :      v ← v'
6 :      j ← index(v')
7 :    else if(v' = S2)
8 :     v ← v'
9 :     j ← index(v')
10 : if(v ≠ null)
11 :  remove_SVA(v)
12 :  for each(v' ∈ v.removed_conflicting_SVAs)
13 :    for each(v" ∈ unassigned_variables)
14 :      if(v' = v")
15 :  v.removed_conflicting _SVAs.clear()
16 :  current_assignmant.remove(j)
17 :  unassigned_variables.add(v)
19 : else
20 :   backtrack
```

Figure5. Dynamic backtracking function.

The table (Figure 7) represents the difference between the three algorithms mentioned in this paper. We introduced only some parts of the algorithm execution, to show the major differences between these techniques. For example, the HS MPP algorithm executed five times each of phase 1 and phase 2 to find the optimal


solution. On the other hand, both HS MPP_BJ and HS MPP_DB algorithms, perform twice the phase 1 and the phase 2.

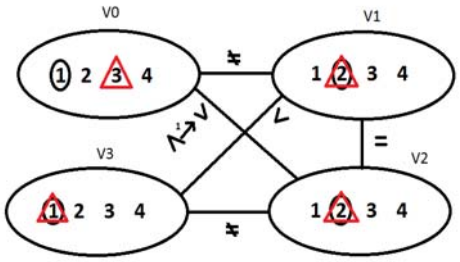


Figure6. Example: problem Θ, assignment a (depicted by circles) and the solution with minimal perturbations (depicted by triangles).

We notice at first, that these last two algorithms are faster than the HS_MPP algorithm.

In addition, in the first implementation of Phase 2, the HS_MPP_BJ removed all the variables of current assignment.

Contrary to the HS_MPP_DB which deleted the variable V0 that caused the failure of the current instantiation in the algorithm. Indeed, the second execution of Phase 1 of these two algorithms, we notice that the HS_MPP_BJ selected three variables (variables deleted in the 1st phase 2 and returned to the 2nd phase 1 in the current assignment. The HS_MPP_DB has selected only one variable. We note secondly that HS_MPP_BJ will consume more time than HS_MPP_DB because variables reductions in the second execution of phase 1.

The optimal solution of the two algorithms is achieved in the second execution of phase 2.

HS MPP	HS MPP_BJ	HS MPP_DB
Phase 1 : CA=V0,V1,V3 UV=V2	**Phase 1 : CA=V0,V1,V3 UV=V2**	**Phase 1 : CA=V0,V1,V3 UV=V2**
Phase 2 : MAC(CA,UV)=null	**Phase 2 MAC(CA,UV)=null**	**Phase 2 : MAC(CA,UV)=null**
Backtrack : CA=V0,V1 UV=V2,V3	**Backtrack : CA=V1,V3 UV=V2**	**Backtrack : CA=0 UV=V2,V0,V1,V3**
Phase 1 : CA=V0,V1 UV=V2,V3	**Phase 1 : CA=V1, V2,V3 UV= V0**	**Phase 1 : CA=V1, V2,V3 UV= V0**
Phase 2 : MAC(CA,UV)= null	**Phase2 : MAC(CA,UV)=V0=3,V1=2,V2=2,V3=1**	**Phase2: MAC(CA,UV)=V0=3,V1=2,V2=2,V3=1**
Backtrack : CA=V0 UV=V2,V3,V1	**ub = 1**	**ub = 1**
Phase 1 : CA=V0,V3 UV=V2,V1		
Phase 2 : MAC(CA,UV)= null		
Backtrack : CA=V0 , UV=V2,V1,V3		
Phase 1 : CA=V0 UV=V2,V1,V3		
Phase 2 : MAC(CA,UV)=null		
Backtrack : CA=null UV=V2,V1,V3,V0		
Phase 1 : CA=V2,V1,V3 UV=V0		
Phase2: MAC(CA,UV)=V0=3,V1=2,V2=2,V3=1		
ub = 1		

The initial solution is depicted by circles and new one by triangles. In this problem one constraint is changed (1). The constraint v0 < v2 was changed to v0 > v2

Figure7. Example of program execution

5. Experimental results

We have carried out a series of experimental tests to compare the original version of HS_MPP to our algorithms: HS_MPP_BJ and HS_MPP_DB.

Experiments were conducted on problems from two different domains-randomly generated CSPs and random instances of the Meeting Scheduling Problem (MSP). In both domains, our algorithms considerably outperformed the original algorithm (HS_MPP).

We evaluate the performance in terms of constraints checking (CCs), computation time in second (Ts) and number of visited nodes (V.n or V.v: visited variables). All this experiments were performed using the Java Programming Language on NetBeans Platform.

5.1 Random

The random CSPs are characterized by < n, d, p1, p2 >, where n is the number of variables, d the number of values for each variable, p1 the density of the constraints network and p2 the constraints tightness. The tests were performed respectively on problems < 20, 10, 0.3, 0.2 > to < 20, 10, 0.3, 0.5>. We injected (added) constraints rate of 1%,3%, 5%,..., 23% and 25%, and for each pair (p1, p2) , 20 instances were solved using each heuristic and the results are presented as an average of

these 20 instances. In these experiments the new solutions of the same problems have the same Hamming distance, i.e. the found solutions are the same.

5.1.1 HS MPP versus HS MPP_BJ

By applying the perturbations to the changed constraints in the range of [1%-25%], we note that the HS MPP_BJ algorithm outperforms the HS MPP algorithm.

Also, by changing the difficulty of the resolved problems, i.e. by varying the value of the tightness P2 (P2 = 0.3 {fig8} P2 =0.4 {fig9}, P2 = 0 5 {fig10}), the HS MPP_BJ algorithm shows significant results than the HS MPP algorithm.

From these results it is clearly apparent that HS MPP_BJ is better than HS MPP through back-jumping research method.

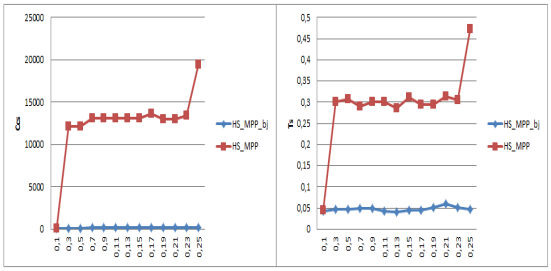


Figure8. Performance of HS_MPP_BJ and HS_MPP on Random Problems (n = 20; p1 = 0,3;p2=0,3) We injected (added) constraints rate of 1%,3%, 5%,..., 23% and 25%. Ccs: constraint checking, Ts: computation time in s.

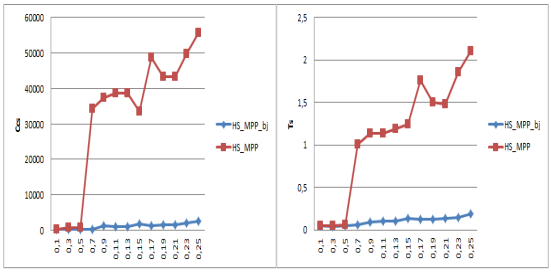


Figure9. Performance of HS_MPP_BJ and HS_MPP on Random Problems (n = 20; p1 = 0,3;p2=0,4) We injected (added) constraints rate of 1%,3%, 5%,..., 23% and 25%.

According to the figures 8, 9 and 10 we can deduce the main difference between HS MPP and HS MPP_BJ algorithms with regard to the number of constraints checks as well as the run time.

5.1.2 HS MPP_DB versus HS MPP_BJ

Using the same parameters in HS_MPP and HS_MPP_BJ comparison, we have compared the two algorithms HS_MPP_BJ and HS_MPP_DB.

We preferred to compare the three algorithms in pairs to show clearly the best results in the figures.

We note in this second comparison, that HS_MPP_DB algorithm surpasses in its turn HS_MPP_BJ algorithm, regarding the amount of tests that increased as well as the computation time because of the re-selected variables in the second execution of Phase 1 by the back-jumping method.

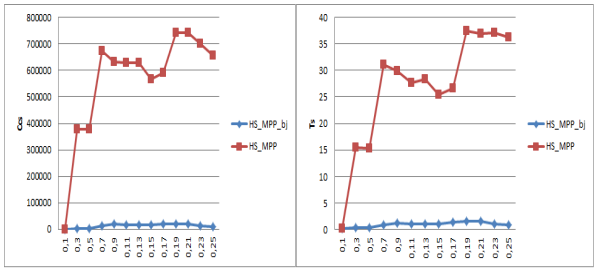


Figure10. Performance of HS_MPP_BJ and HS_MPP on Random Problems (n = 20; p1 = 0,3;p2=0,5) We injected (added) constraints rate of 1%,3%, 5%,..., 23% and 25%.

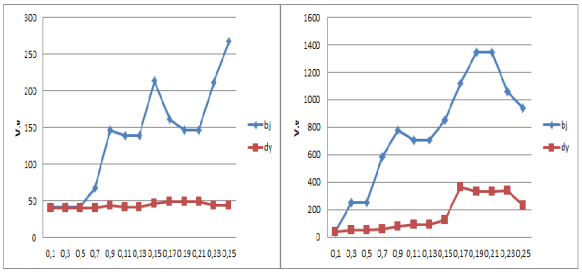


Figure11. Performance of HS_MPP_BJ and (HS_MPP_DB or Dy) on Random Problems (n = 20; p1 = 0,3; p2=0,4 and 0,5) V.v : visited variables.

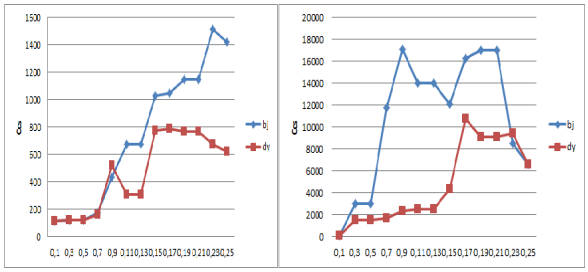


Figure12. Performance of HS_MPP_BJ and (HS_MPP_DB or Dy) on Random Problems (n = 20; p1 = 0,3; p2=0,4 and 0,5).

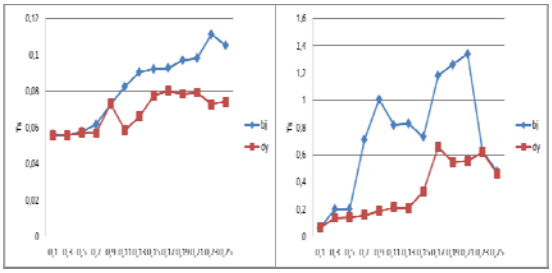


Figure13. Performance of HS_MPP_ BJ and (HS_MPP_DB or Dy) on Random Problems (n = 20; p1 = 0,3; p2=0,4 and 0,5).

Finally, we can note that the intelligent backward reasoning technique, especially, dynamic backtracking, is very relevant in the context of repairing and minimal

perturbation problems. It shows that a new solution of the changed problem can be found rapidly within minimal perturbations.

5.2 Meeting scheduling problems

Many CSP researchers use random uniform instances to evaluate their constraint satisfaction algorithms. Although it is generally agreed that the ultimate test of a CSP algorithm is its performance on "real world". For this purpose, we have evaluated our algorithms using the Meeting Scheduling Problems (MSPs).

Meetings are an important vehicle for human communication. The Meeting Scheduling Problem consists of a set of people which use their personal calendars to determine when and where one or more meeting (s) could take place [17].

The MSP is characterized by <m, p, n, d, h, t, a>, where:

m: is the number of meetings;

p: is the number of participants;

n: is the number of meetings per participant;

d: is the number of days;

h: is the number of hours per day;

t: is a duration of the meeting;

a: is the percentage of availability for each participant.

We present our results for the class <20, 5, 15, 5, 10, 1, 70> and we vary the rate of changed constraints in the range of [1%-25%]. We generated 20 different instances were solved using each heuristic and the results are presented as an average of these 20 instances. In these experiments the new solutions of the same problems have the same Hamming distance, i.e. the found solutions are the same.

5.2.1 HS MPP versus HS MPP_BJ versus HS MPP_DB

Figures 14, 15 and 16 depict the results of comparing performance in terms of Ccs of HS_MPP to our two algorithms. The number of constraints checking in our HS_MPP_BJ and HS_MPP_DB algorithms was reduced compared to that of HS_MPP. In fact, computing time will be also reduced. These results demonstrate the robustness of our algorithms.

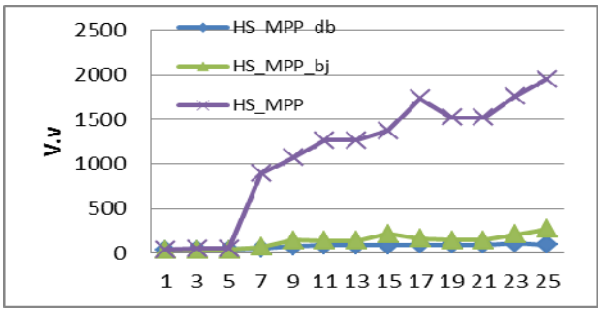


Figure14. Performance of HS_MPP vs HS_MPP_BJ vs (HS_MPP_DB or Dy) on Meeting scheduling Problems V.v: Visited variables.

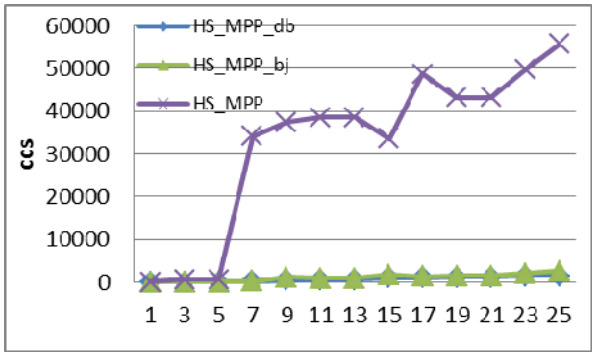


Figure15. Performance of HS_MPP vs HS_MPP_BJ vs (HS_MPP_DB or Dy) on Meeting scheduling Problems.

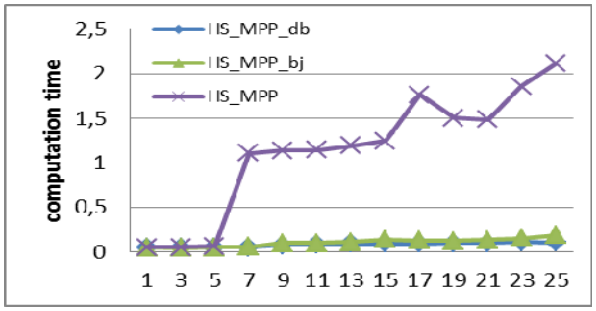


Figure16. Performance of HS_MPP vs HS_MPP_BJ vs (HS_MPP_DB or Dy) on Meeting scheduling Problems.

5.2.2 HS MPP_DB versus HS MPP_BJ

Similarly to the results obtained previously with random problems, we can deduce that HS_MPP_DB algorithm outperforms the HS_MPP_BJ (figures 17, 18 and 19), concerning the amount of tests that increased as well as the computation time because of the re-selected variables in the second execution of Phase 1 by the back-jumping method.

Based on these results we can affirm that the dynamic backtracking is very relevant in the context of repairing and minimal perturbation problems. It shows that a new solution of the changed problem can be found rapidly within minimal perturbations.

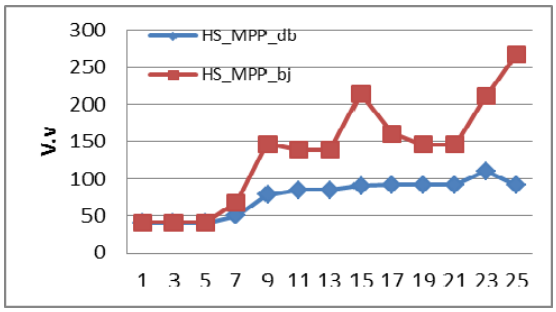


Figure17. Performance of HS_MPP_BJ vs (HS_MPP_DB or Dy) on Meeting scheduling Problems.

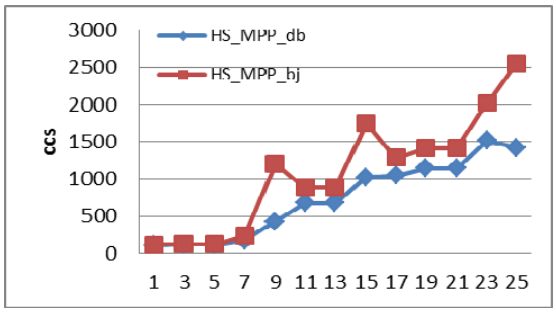


Figure18. Performance of HS_MPP_BJ vs (HS_MPP_DB or Dy) on Meeting scheduling Problems.

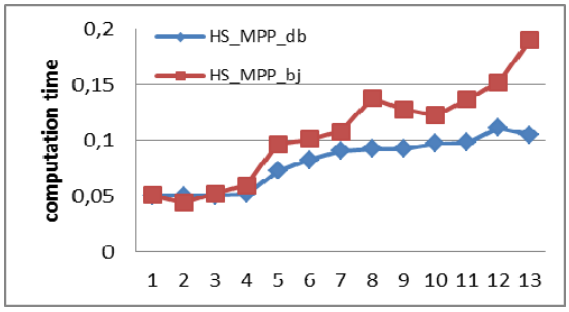


Figure19. Performance of HS_MPP_BJ vs (HS_MPP_DB or Dy) on Meeting scheduling Problems.

6 Conclusion

In this paper, we proposed two new approaches, namely HS MPP_BJ and HS MPP_DB. These approaches are inspired by HS_MPP algorithm of repair and search for the optimal solution.

HS_MPP_BJ is based on the back-jumping, it allows directly the deletion of the culprit variable which prevent the algorithm to extend partial solution, as well as the variables that follow it in the current assignment of phase 2.

HS_MPP_DB is based on an intelligent dynamic backtracking technique that just removes the culprit variable of the current assignment.

In general, these last two algorithms outperform HS_MPP. But HS_MPP_DB remains the best of the three algorithms through dynamic backtracking which memorize the causes of failures.

The obtained results encourage us to investigate our approaches in other real problems in artificial intelligence such as: assigning nurses to shifts, breakdown of machinery and delayed flights.

References

[1] Zivan Roie, Alon Grubshtien, Amnon Mei-sels. 2011 Hybrid search for minimal perturbation in Dynamic CSPs. Constraint 16, no. 3 : 228-249 .

[2] El Sakkout H, Richards T, Wallace M. Minimal perturbation in dynamic scheduling. In: Prade Proceedings of the 13th European Conference on Artificial Intelligence, ECAI-98, 1998.

[3] Bartak R, Muller T, Rudova H. A new approach to modeling and solving minimal perturbation problems. In: Apt KR, Fages F, Rossi F, Szeredi P, Vncza J Recent Advances in Constraints, Lecture Notes in Computer Science, vol 3010, Springer Berlin, pp 233-249, 2004.

[4] Rudova H, Muller T, Murray K. Complex university course timetabling. Journal of Scheduling 14(2):187-207, 2011.

[5] Bessiere C and Regin J.C. MAC and combined heuristics: two reasons to forsake FC (and CBJ?) on hard problems. In Proc. Second International Conference on Principles and Practice of Constraint Programming, CP 96, pages 6175, Cambridge MA, 1996.

[6] Gaschnig J. A general backtracking algorithm that eliminates most redundant tests. In Proceedings of the International Joint Conference on Artificial Intelligence, pages 457–1977, 1979.

[7] Maestre A, Bessière C, and Meseguer P. Dynamic backtracking distribué. In 7 èmes Journées nationales sur la résolution pratique de problèmes NP Complets, pages 61–72, Toulouse, France, 2001.

[8] R. Dechter, A. Dechter. Belief maintenance in dynamic constraint networks. In Proceedings of AAAI-88, pp. 37-42, 1988.

[9] N. Roos, Y. Ran, H. J. van den Herik. Combining local search and constraint propagation to find a minimal change solution for a dynamic csp. In Artificial intelligence: Methodology, systems, applications (pp. 272–282), 2000.

[10] N. Roos, Y. Ran, H. J. van den Herik, Approaches to find a near-minimal change solution for dynamic csps. In Fourth international workshop on integration of AI and OR techniques in constraint programming for combinatorial optimisation problems (pp. 373–387), 2002.

[11] E. Hebrard, B. Hnich, B. O'Sullivan, T. Walsh. diverse and similar solutions in constraint programming. In The twentieth national conference on artificial intelligence, AAAI-2005. Pittsburgh, PA, USA, 2005.

[12] E. Hebrard, B. O'Sullivan, T. Walsh. Distance constraints in constraint satisfaction. In The twentieth international joint conference on

artificial intelligence, IJCAI-2007. Hyderabab, India, 2007.
[13] Y. Acodad, I. Benelallam, S. Hammoujan, E.H. Bouyakhf. Extended Partial-order Dynamic Backtracking algorithm for dynamically changed environments. In The 24th IEEE International Conference on Tools with Artificial Intelligence ICTAI-2012.
[14] M. L. Ginsberg, D. A. McAllester.GSAT and Dynamic Backtracking. Journal of Artificial Intelligence Research, 25-46, 1994.
[15] MOHAMED, Azlinah, YUSOFF, Marina, MOHTAR, Itaza Afiani, et al. Constraint satisfaction problem using modified branch and bound algorithm. WSEAS Transactions on Computers, 2008, vol. 7, no 1, p. 1-7.
[16] MOHAMED, Azlinah, YUSOFF, Marina, MUTALIB, Sofianita, et al. Modified branch and bound algorithm. In : Proceedings of the 8th Conference on 8th WSEAS International Conference on Evolutionary Computing-Volume 8. World Scientific and Engineering Academy and Society (WSEAS), 2007. p. 274-279.
[17] Demirel, I. O., & Erdogan, N. (2007, February). Meeting scheduling with multi agent systems: design and implementation. In SEPADS'07: Proceedings of the 6th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (pp. 92-97).

Biographies

El Mehdi El Graoui received his M.Sc. in computer science and telecommunications from Mohammed V University of Rabat, faculty of Science, Morocco in 2012. He is a PhD student at LIMIARF Laboratory under the supervision of Mr. El Houssine BOUYAKHF in Mohammed V University of Rabat. He has published papers in various international conferences. His research interests include the satisfaction and optimization of constraints problems and the artificial intelligence.

Imade BENELALLAM is currently an Assistant Professor teaching at the National Institute of Statistics and applied Economic. He works also within the LIMIARF Laboratory Mohammed V University-Agdal. Imade BENELALLAM received his Ph.D. degree in Computer Science from Mohammed V University-Agdal Morocco in April 2010. He did his Ph.D. under the supervision of professor El Houssine Bouyakhf director of LIMIARF Lab with the collaboration of professor Christian Bessiere director of research at CNRS, University Montpellier 2, France. His Ph.D. research focused on Distributed Constraint Reasoning. The title of his thesis is: "Exact approaches to DisCSPs and DCOPs problems''. From 2005 to 2010 he was a computer science engineer at Mohammed V University-Agdal Morocco. Now he is a member of the AI team. AI is a research team that aims to propose and combine models and algorithms in constraints, learning and agents. He works in different projects with industrial partner; the most significant was with Thales group. Imade BENELALLAM is also the Chair of the GOLD Affinity Group of IEEE Morocco section. He was the IEEE GOLD representative of Morocco section in various congresses.

El Houssine BOUYAKHF is full Professor at the Faculty of Sciences, Mohammed-V University, Rabat, teaching Computer Sciences, Pattern Recognition, Image Processing and Artificial Intelligence. He is the scientific leader of LIMIARF Lab (Laboratory of Informatics, Applied mathematics, Artificial Intelligence and Pattern recognition). He received the Engineer degree from Sup'Aéro (ENSAE) National Higher School of Aeronautics and Space, Toulouse, France; he received the Doctor Engineer degree in Pattern recognition and Artificial Intelligence from University Paul Sabatier, Toulouse, France and "Doctorat d'Etat" in Robotics and Artificial Intelligence from LAAS of CNRS and University Paul Sabatier, Toulouse, France. His main topics of interest are: Artificial Intelligence and Constraint programming, Robotics and Vision, and Telecommunications. El Houssine BOUYAKHF supervises several PhD theses in the research themes listed before and site leader or key person of international projects. He has more than 100 scientific publications.

www.icgst.com

Evolving Stock Market Prediction Models Using Multi-gene Symbolic Regression Genetic Programming

Alaa F. Sheta[1], Sara Elsir M. Ahmed[2], Hossam Faris[3]
[1] *Computers and Systems Department, Electronics Research Institute, Giza, Egypt*
[2] *Computer Science Department, Sudan University of Science and Technology, Sudan*
[3] *Business Information Technology Department, The University of Jordan, Amman, Jordan*
asheta66@gmail.com, saraelsir1@yahoo.com, hossam.faris@ju.edu.jo

Abstract

Stock market is a dynamic, non-linear, complex, and chaotic process in nature. Predicting stock market is an important financial problem which receives increasing attention in the past few decades. The main objective of this paper is to build a suitable prediction model for the Standard & Poor 500 return index (S&P500) with potential influence feature using multi-gene Symbolic Regression genetic programming (GP). The experiments and analysis developed in this research show many advantages for the multi-gene GP. Multi-gene GP evolves linear combinations of non-linear functions of the input variables. A comparison between traditional multiple linear regression model and multi-gene GP is provided. Multi-gene GP shows more robust results especially in the validation/testing case.

Keywords: *Stock market prediction, S&P500, Genetic Programming, Multi-gene Symbolic Regression.*

1 Introduction

The World of Finance is growing, and stock market exchange constitutes an excessively large amount of this thriving. Stock Market forecasting is considered as one of the most stimulating tasks today. Great attention was dedicated to understanding, analyzing and forecasting future stock prices and developing financial time series for such purposes. Many aspects affect the stock market prices which include business cycles, interest rates, monitory policies, general economic conditions, traders' expectations, political events, etc. [1]. One essential aspect of stock market price forecasting is developing models which should be able to predict the correct value of the future stock market price index. Existing time series forecasting methods commonly drop into two groups according to [2]: 1) classical methods which are grounded on statistical/mathematical notions, and 2) modern heuristic methods which are based on algorithms from the field of artificial intelligence.

Evolutionary Computation (EC) fall within the AI search techniques. EC adopts Darwinian notion of natural genetic to solve complex optimization problems. EC are population based approach. This population is evolved based on a guided random search and set of evolutionary parameters to explore the search space of a problem. EC techniques include: Genetic Algorithms [3], Genetic Programming [4], Evolutionary Strategies [5], Differential Evolution [6, 7], Particle Swarm Optimization [8] and many others [9]. This paper is considered as part of the modern heuristic methods.

1.1 Literature Review

EC technique can automatically solves problems without requiring any *a prior* knowledge about the system structure or system function characteristics in advance. Since early nineties, GP has been used to solve a wide variety of real-world problems, creating better results than human and even better than other problem solver for optimization. GP evolved rapidly [10], with new concepts, methods and real-world problem solving. GP was able to outperform other traditional techniques in solving modeling and identification problems. GP provided solution to many problems in classification [11, 12], manufacture process modeling [13, 14, 15], marketing problems [16], and stock market forecasting [17].

Most current GP implementations are restricted to a single gene structure which is used to develop a relationship between a system/model inputs and outputs [18, 10, 19]. Recently, multi-gene genetic programming was presented to better evolve a mathematical model for nonlinear system dynamics [20]. A multi-gene individual always consists of one or more genes,

each of which is a regular GP tree. Genes in multigene are learned in an incremental manner to generate individuals such that an enhanced fitness is achieved. The produced multi-gene GP model is a weighted linear combination of each gene.

In [21], author built a technical trading system with genetic programming to test the efficiency of Chinese stock markets. The proposed system used historical prices and volumes as main inputs for the forecasting system where randomly generated trading rules composed of basic functions were optimized using genetic programming. It was reported that the optimal technical trading rules generated based GP have statistically significant out-of-sample excess returns compared with the well-known buy-and-hold strategy.

A prediction model for the S&P500 index based GP was presented in [17]. The developed experiments show some unique advantages of using GP compared to linear regression and fuzzy logic in stock market modeling. Such advantages include generating mathematical models, which are simple to evaluate and having powerful variable selection mechanism that identifies significant variables. A possibly profitable trading strategy was proposed in [22] using GP. GP was used to evolve regression models which produce reasonable one-day-ahead forecasts. An experimental study for the Egyptian Stock Index was presented in [23]. Author showed that GP has the capability to provide accurate prediction for the stock market when compared to traditional machine learning algorithms such as ANN.

The paper is organized as follows: Section 2 provide a brief overview on the modeling based regression. Section 3 describes the main features of Genetic Programming and the tree representation of the problem. In Section 4, we emphasizes on the symbolic regression method and show its main characteristics. The fitness function and performance evaluation criterion are given in Section 5. The details of the S&P 500 data set and the 27 potential financial and economic variables that affect the stock movement are described in Section 6. The developed regression and GP models along with the performance evaluation criterion are given in Section 7. Finally, we provide a conclusion of this work.

2 Regression Model

In this section, we describe the mathematical equation which govern for multiple-linear regression model. The general equation is given:

$$\hat{y} = \alpha_0 + \sum_{i=0}^{n} \alpha_i U_i \tag{1}$$

where U_i represents the model input variables ($i = 1, \ldots, 27$) in our case. $\hat{y}$ is the model output variable, which is the stock index. To show how the parameter estimation process work, we assume that the model mathematical equation can be expanded as:

$$y = \alpha_0 + \alpha_1 U_1 + \cdots + \alpha_{27} U_{27} \tag{2}$$

To find the values of the model parameters α's we need to build what is called the regression matrix ψ. This matrix is developed based on the experiment collected measurements. Thus, ψ can be presented as follows given there is a set of measurements m:

$$\Phi = \begin{pmatrix} 1 & U_1^1 & U_2^1 & \ldots & U_{27}^1 \\ 1 & U_1^2 & U_2^2 & \ldots & U_{27}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & U_1^m & U_2^m & \ldots & U_{27}^m \end{pmatrix}$$

The parameter vector θ and the output vector y can be presented as follows:

$$\theta = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{27} \end{pmatrix} \qquad y = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{pmatrix}$$

Least squares estimation solution yields the normal equation:

$$\Phi^T \theta = y \tag{3}$$

It has a solution:

$$\theta = \Phi^{-1} y \tag{4}$$

But since, the regression matrix Φ is not a symmetric matrix, we have to reformulate the equation such that the solution for the parameter vector θ is as follows:

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T y \tag{5}$$

3 Genetic Programming

GP is an evolutionary algorithm which is inspired by the principles of Darwinian evolution theory and natural selection [10]. GP is domain-independent modeling technique used to create mathematical models based on data sets that describes complex problems or processes [18]. GP it is commonly referred to as Symbolic Regression. The concept of Symbolic Regression was first introduced by John Koza in [18].

3.1 How GP Works?

GP algorithms works iteratively as an evolutionary cycle. Through this cycle, GP evolves a population of computer programs or models represented as symbolic tree expressions to solve complex optimization problem. Traditionally the evolved models are LISP programs. Since GP automatically evolves both the structure and the parameters of the mathematical model, LISP gives GP more flexibility to handle


data and structures that can be easily manipulated and evaluated. For example, the simple expression: $(sin(X) + (\frac{Y}{5}))$ is represented as shown in Figure 1.

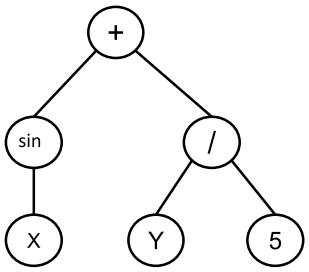


Figure 1: Example of basic tree representation in GP.

In Figure 2, we show the evolutionary process of GP. In more details, the cycle is described as follows:

- **Initialization:** the GP cycle starts by generating an initial population of random computer programs (also known as individuals) using a predefined function set and a terminal set.
- **Fitness evaluation:** the fitness value for each individual is computed based on a defined measurement.
- **Selection:** based on the fitness values of the individual, some of these individuals are chosen for reproduction. Selection is done using some selection mechanism (i.e; Tournament selection).
- **Reproduction:** in this process, different reproduction operators are applied in order to generate new individuals. These operators usually include crossover, mutation and elitism. Crossover operator swaps two randomly chosen sub-parts in two randomly chosen individuals. Mutation operator selects a random point in an individual and replaces the part under this point with a new generated sub-part. Elitism selects some best individuals and copies them to next generation without any modification.
- **Termination:** the evolutionary cycle of the GP algorithm stops iterating when an individual with a required fitness value is found or the predefined maximum number of iterations is reached.

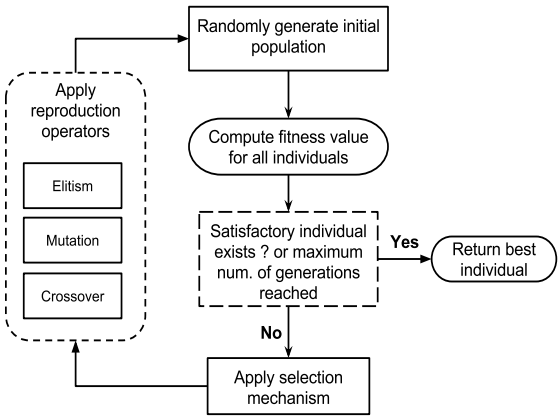


Figure 2: Main loop of the GP [24].

4 Multigene Symbolic Regression GP

Multigene symbolic regression is a special variation of the classic GP algorithms where each symbolic model is represented by number of GP trees weighted by linear combination [20]. Each tree is considered as a "gene" by itself. The prediction of the output variable $\hat{y}$ is formed by combining the weighted outputs the trees/genes in the Multigene individual plus a bias term. Each tree is a function of zero or more of the N input variables $u_1, \ldots, u_N$. Mathematically, a Multigene regression model can be written as:

$$\hat{y} = \gamma_0 + \gamma_1 \times Tree_1 + \cdots + \gamma_M \times Tree_M \quad (6)$$

where γ_0 represents the bias or offset term while $\gamma_1, \ldots, \gamma_M$ are the gene weights and M is the number of genes (i.e. trees) which constitute the available individual. The weights (i.e. regression coefficients) are automatically determined by a least squares procedure for each multi-gene individual. An example of multi-gene model is shown in Figure 3. The presented model can be introduced mathematically as given in Equation 7.

$$\gamma_0 + \gamma_1(sin(X) + (\frac{Y}{5})) + \gamma_2(5 * Z) \quad (7)$$

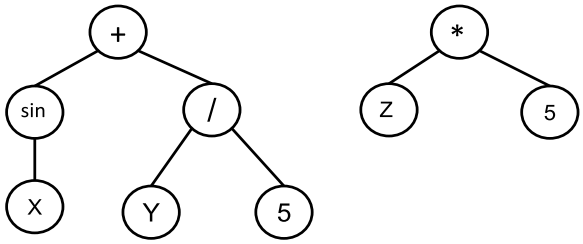


Figure 3: Example of Multi-gene GP model

In general, symbolic regression has many advantages over other identification models. GP enjoys high flexibility since it searches both the space of models along with the space of all possible parameters simultaneously. Moreover, the final generated models are usually compact mathematical models and can be easily assisted and evaluated. Therefore, GP models are

considered to be more interpretable compared to other identification and modeling approaches like Artificial Neural Network. In the case of multi-gene GP in particular, the generated models have the advantage of combining between the classical linear regression and the ability to represent non-linear behaviors [20].

5 Fitness Function

Fitness function is essential for any evolutionary computation process. In the case of GP, we adopted the Mean absolute error (MAE) as a fitness function as given in Equation 8.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}| \tag{8}$$

In order to check the performance of the developed GP model, we also explored other performance evaluation functions such as: the Root mean square error (RMSE), Relative absolute error (RAE) and Root relative squared error (RRSE). These performance evaluation functions are used to measure how close the computed stock index values computed by the GP model to the real index measured values. The equations which describe the computed criterion are presented as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2} \tag{10}$$

$$RAE = \frac{\sum_{i=1}^{n}|y - \hat{y}|}{\sum_{i=1}^{n}|y - \overline{y}|} \tag{11}$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(y - \hat{y})^2}{\sum_{i=1}^{n}(y - \overline{y})^2}} \tag{12}$$

where y is the actual stock index value, $\hat{y}$ is the estimated stock index value and $\overline{y}$ is the mean of the signal y using n measurements.

We also compute the correlation coefficient for the two signals y and $\hat{y}$. The correlation coefficient of two signals is the covariance between the two signals divided by the product of their individual standard deviations. It is a normalized measurement of how the two are linearly related.

$$R_{xy} = \frac{S_{xy}}{S_x S_y} \tag{13}$$

Formally, the sample correlation coefficient is as given in Equation 13, where S_x and S_y are the sample standard deviations, and S_{xy} is the sample covariance.

6 S&P 500 Data Set

In this work, we use 27 potential financial and economic variables that impact the stock movement. The main consideration for selecting the potential variables is whether they have significant influence on the direction of (S&P 500) index in the next week. While some of these features were used in previous studies [25]. We can categorize these variables into six groups as follows:

G_1: S&P 500 index return in three previous days SPY(t-1), SPY(t-2), SPY(t-3).

G_2: Financial and economical indicators (Oil, Gold, CTB3M, AAA).

G_3: The return of the five biggest companies in S&P 500 (XOM, GE, MSFT, PG, JNJ).

G_4: Exchange rate between USD and three other currencies (USD-Y, USD-GBP, USD-CAD).

G_5: The return of the four world major indices (HIS, FCHI, FTSE, GDAXI).

G_6: S&P 500 trading volume (V).

S&P 500 stock market data set used in our case consists of 27 features, which cover five-year period starting 7 December 2009 to 2 September 2014. The S&P 500 data were presented and sampled on a weekly basis such that only 143 samples constitute the whole data set. 100 samples were used as training set and 43 samples were used for testing the developed model. The list, the description, and the sources of the potential features are given in Table 1.

7 Experimental Results

In this section, we provide two types of experiments to develop two models: a multi-linear regression model and a multi-gene GP model. The mathematical equations which describe the models, the training and testing graphs of the data and the evaluated performance criterion shall be provided in the following sections.

7.1 Regression Model

The values of the parameters α is estimated using LSE method to produce the values of the parameters α_i given in Equation 1 and Equation 2. The produced linear regression model can be presented as given in Table 2. The actual and Estimated S&P 500 index values based the MLR in both training and testing cases are shown in Figure 4.

7.2 GP Model

To develop our multi-gene GP model, some parameters have to be defined by the user at the beginning of


Table 1: The 27 potential influential features of the S&P 500 Index [25]

Variable	Feature	Description
x_1	SPY(t-1)	The return of the S&P 500 index in day $t-1$ Source data: finance.yahoo.com
x_2	SPY(t-2)	The return of the S&P 500 index in day $t-2$ Source data: finance.yahoo.com
x_3	SPY(t-3)	The return of the S&P 500 index in day $t-3$ Source data: finance.yahoo.com
x_4	Oil	Relative change in the price of the crude oil Source data: finance.yahoo.com
x_5	Gold	Relative change in the gold price Source data: www.usagold.com
x_6	CTB3M	Change in the market yield on US Treasury securities at 3-month constant maturity, quoted on investment basis Source data: H.15 Release - Federal Reserve Board of Governors
x_7	AAA	Change in the Moody's yield on seasoned corporate bonds - all industries, Aaa Source data: H.15 Release - Federal Reserve Board of Governors
x_8	XOM	Exxon Mobil stock return in day t-1 Source data: finance.yahoo.com
x_9	GE	General Electric stock return in day t-1 Source data: finance.yahoo.com
x_{10}	MSFT	Micro Soft stock return in day t-1 Source data: finance.yahoo.com
x_{11}	PG	Procter and Gamble stock return in day t-1 Source data: finance.yahoo.com
x_{12}	JNJ	Johnson and Johnson stock return in day t-1 Source data: finance.yahoo.com
x_{13}	USD-Y	Relative change in the exchange rate between US dollar and Japanese yen Source data: OANDA.com
x_{14}	USD-GBP	Relative change in the exchange rate between US dollar and British pound Source data: OANDA.com
x_{15}	USD-CAD	Relative change in the exchange rate between US dollar and Canadian dollar Source data: OANDA.com
x_{16}	HIS	Hang Seng index return in day t-1 Source data: finance.yahoo.com
x_{17}	FCHI	CAC 40 index return in day t-1 Source data: finance.yahoo.com
x_{18}	FTSE	FTSE 100 index return in day t-1 Source data: finance.yahoo.com
x_{19}	GDAXI	DAX index return in day t-1 Source data: finance.yahoo.com
x_{20}	V	Relative change in the trading volume of S&P 500 index Source data: finance.yahoo.com
x_{21}	CTB6M	Change in the market yield on US Treasury securities at 6-month constant maturity, quoted on investment basis Source data: H.15 Release - Federal Reserve Board of Governors
x_{22}	CTB1Y	Change in the market yield on US Treasury securities at 1-year constant maturity, quoted on investment basis Source data: H.15 Release - Federal Reserve Board of Governors
x_{23}	CTB5Y	Change in the market yield on US Treasury securities at 5-year constant maturity, quoted on investment basis Source data: H.15 Release - Federal Reserve Board of Governors
x_{24}	CTB10Y	Change in the market yield on US Treasury securities at 10-year constant maturity, quoted on investment basis Source data: H.15 Release - Federal Reserve Board of Governors
x_{25}	BBB	Change in the Moody's yield on seasoned corporate bonds - all industries, Baa Source data: H.15 Release - Federal Reserve Board of Governors
x_{26}	DJI	Dow Jones Industrial Average index return in day t-1 Source data: finance.yahoo.com
x_{27}	IXIC	NASDAQ composite index return in day t-1 Source data: finance.yahoo.com

Table 2: A Regression Model with Inputs: $x_1, \ldots, x_{27}$

$$
\begin{aligned}
\hat{y} &= -0.0234 * x_1 + 0.13 * x_2 + 0.021 * x_3 + 0.021 * x_4 - 0.021 * x_5 \\
&- 10.303 * x_6 + 6.0031 * x_7 + 0.7738 * x_8 + 0.2779 * x_9 - 0.43916 * x_{10} \\
&- 0.27754 * x_{11} + 0.12733 * x_{12} - 0.058638 * x_{13} + 13.646 * x_{14} + 9.5224 * x_{15} \\
&- 0.0003 * x_{16} + 0.24856 * x_{17} - 0.0016 * x_{18} + 0 * x_{19} - 2.334 \times 10^{-9} * x_{20} \\
&+ 0.16257 * x_{21} + 0.63767 * x_{22} - 0.14301 * x_{23} + 0.08 * x_{24} + 0.074 * x_{25} \\
&- 0.0002 * x_{26} + 0.026301 * x_{27} + 6.9312 \qquad (14)
\end{aligned}
$$

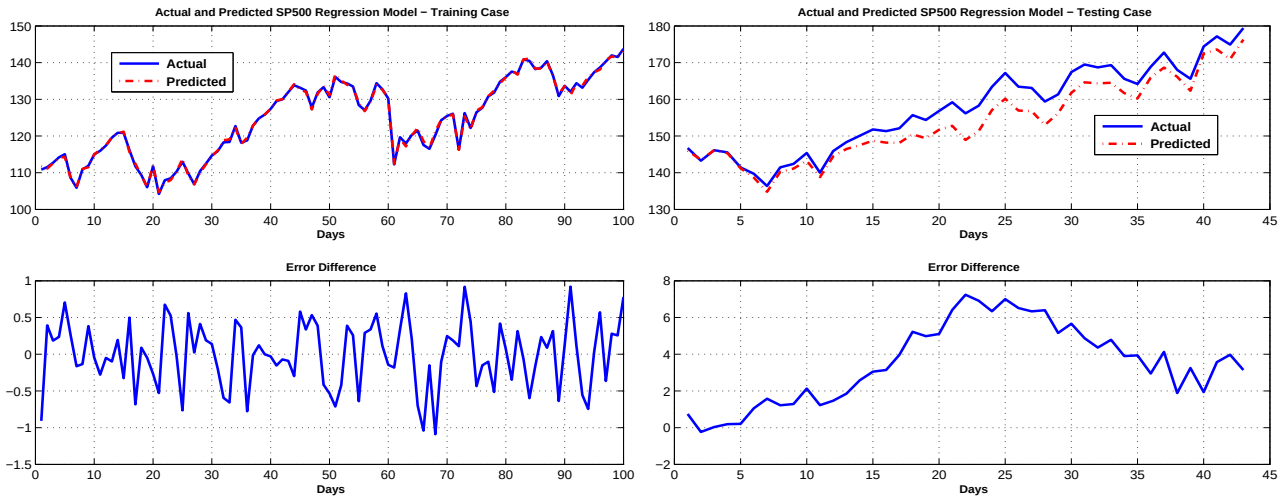



Figure 4: Regression: Actual and Estimated S&P 500 Index values and Error difference between the two curves (a) Training Case (b) Testing Case

the evolutionary process. These parameters include the population size, selection mechanism, crossover and mutation probabilities, the maximum number of genes allowed to constitute the multi-gene and many others. User has to setup the maximum number of genes G_{max} where a model is allowed to have. These setup parameters are presented in Table 3. The maximum tree depth D_{max} allows us to change the complexity of the evolved models. Restricting the tree depth helps evolving simple model but it may also reduce the performance of the evolved model. The adopted function set to develop the GP model is given as:

$$F = \{+, -, \times\}$$

Table 3: GP Tuning Parameters

Population size	30
Number of generations	300
Selection mechanism	Tournament
Tournament size	10
Max. tree depth	10
Probability of crossover	0.85
Probability of mutation	0.1
Number of inputs	27
Max. genes	7
Function set	+, -, ×
Constants range	[-10 10]

7.2.1 GPTIPS Toolbox

In this research, we adopted a MATLAB software toolbox called Genetic Programming & Symbolic Regression for MATLAB (GPTIPS) toolbox [20] to develop our results. GPTIPS is an open source genetic programming toolbox for multi-gene symbolic regression. This software tool offers a number of suitable functions for exploring the population of possible models, thus examining model behavior, post-run a model simplification function and export the model to number of formats, such as graphics file, LaTex expression, symbolic math object or standalone MATLAB file [20].

One of the main characteristics of GPTIPS is that it can be configured to evolve multi-gene individuals. Some features can be summarized as follows [20]:

- Multiple tree (multi-gene) individuals.
- Tournament selection & lexicographic tournament selection [26].
- Standard sub-tree crossover operator.
- Elitism.
- Early run termination criterion.
- Graphical population browser showing best and non-dominated individuals (fitness & complexity).
- Graphical summary of fitness over GP run.
- 6 different mutation operators.

In Table 4, we show the mathematical equation evolved for index prediction using multi-gene GP. It can be clearly seen that the final model is a simple and compact mathematical model which is easy to evaluate. Figure 5 shows the actual and estimated stock market values based the developed GP model in both training and testing cases. In Figure 6, we show the convergence of GP over 300 generations. The performance measurements for the model were computed and summarized in Table 5.


Table 4: A GP model with Inputs: $x_1, \ldots, x_{27}$

$$
\begin{aligned}
\hat{y} &= 0.2206 * x_1 - 0.3617 * x_5 - 6.6983 * x_6 + 32.817 * x_{14} + 0.8029 * x_{15} + 0.382 * x_{22} \\
&- 0.0556 * x_{25} + 0.085 * x_{27} - 0.1 * x_1 * x_{14} + 0.4542 * x_5 * x_{14} - 0.1 * x_6 * x_{14} \\
&+ 0.51 * x_{11} * x_{15} + 0.3617 * x_{14} * x_{21} + 0.1325 * x_{14} * x_{22} + 0.302 * x_{14} * x_{25} \\
&- 0.1 * x_{14} * x_{27} - 0.1 * x_{14}^2 * x_{21} + 104.35 * x_{14}^2 + 0.1 * x_5 * x_{14} * x_{15} \\
&+ 0.1 * x_{11} * x_{14} * x_{15} - 55.494 \qquad (15)
\end{aligned}
$$

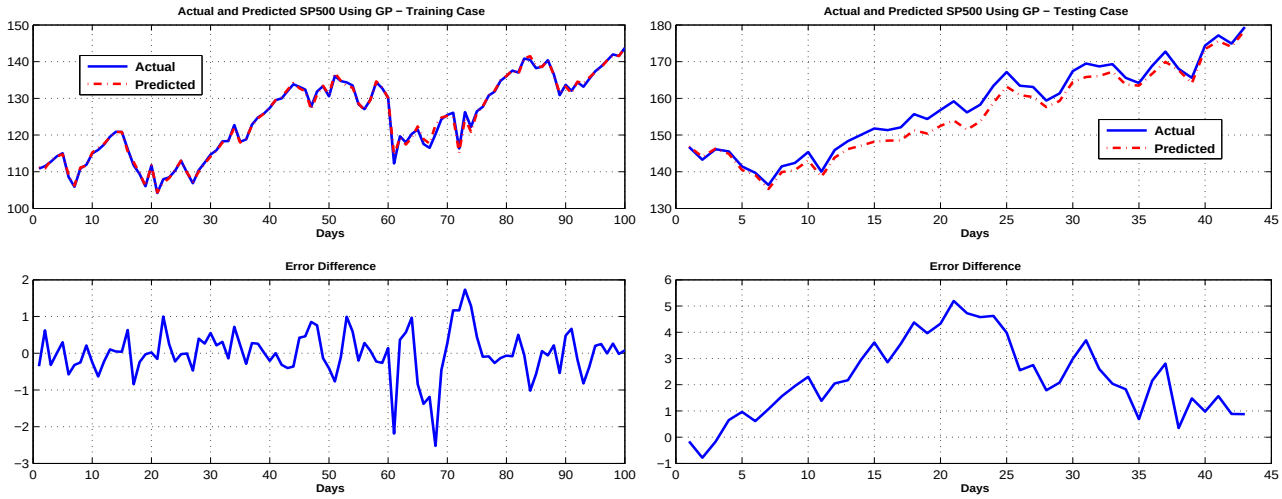


Figure 5: Multi-gene GP: Actual and Estimated S&P 500 Index values and Error difference between the two curves (a) Training Case (b) Testing Case

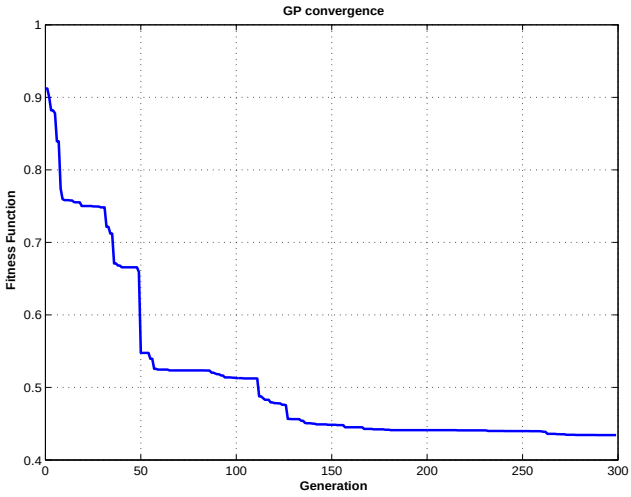


Figure 6: Convergence of GP

8 Comments on the Results

As given in Table 5, the computed MAE in training case for the regression model is better than that of the GP model. Meanwhile, in the testing case, the GP model is more stable in the prediction case and provided a better MAE. This is also true for other adopted criteria. From Figure 5 and Figure 4, the characteristics of the testing case for the GP looks better since the actual and estimated curves are closer than in the case of multiple regression model. In Figure 7 (a) and (b), we also show the scattered plot for the multiple regression model and the Multi-gene GP models.

9 Conclusion

In this paper, we explored the use of Multi-gene symbolic regression GP model to develop a prediction model for the S&P500 stock index. A comparison between the proposed model and traditional multiple linear regression model was presented. We used 27 potential financial and economic variables which impact the stock movement. The main consideration for selecting the potential variables was based on their

Table 5: Evaluation Criteria for the developed models

Criterion	Regression		Multi-gene GP	
	Training	Testing	Training	Testing
Correlation coefficient	0.999	0.985	0.998	0.992
Mean absolute error	0.360	3.533	0.434	2.294
Root mean squared error	0.444	4.120	0.582	2.677
Relative absolute error	3.889%	35.119%	0.621%	22.808%
Root relative squared error	4.232%	35.580%	5.917%	23.122%
Total Number of Instances	100	43	100	43

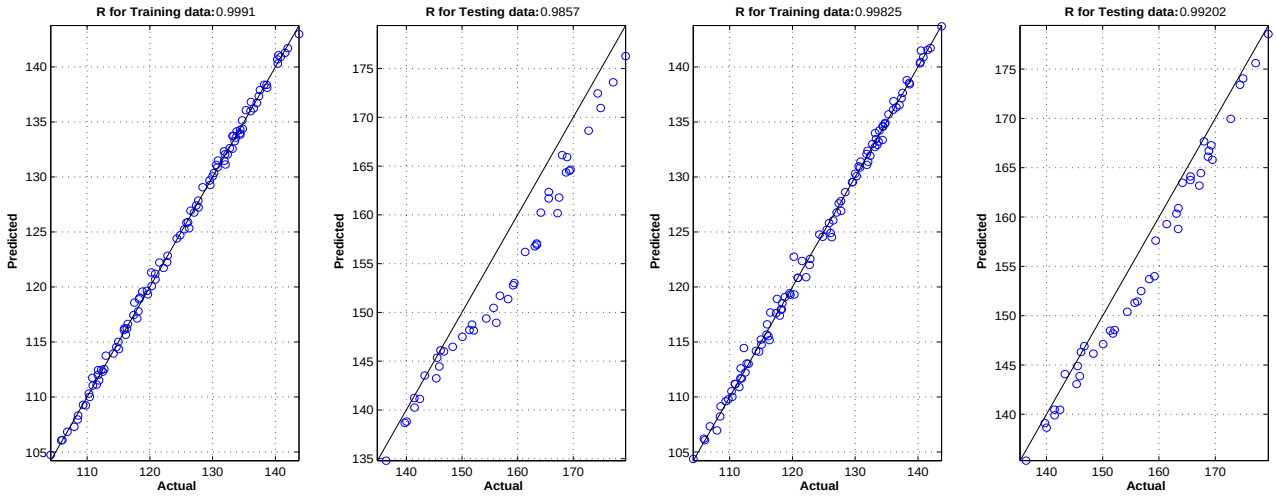


Figure 7: (a) Regression Scattered Plot (b) GP Scattered Plot

significant influence on the direction of S&P 500 index. The data set was sampled on a weekly bases. The developed GP model provided good prediction capabilities especially in the testing case. The results were validated using number of evaluation criteria. The knowledge gained is comprehensible and can enhance the decision making process. Future research shall focus on exploring other advantages of GP, which is the capability of identifying the most important variables in such a dynamic and complex problem.

References

[1] Mahesh Khadka, Benjamin Popp, Kayikkalthop M. George, and Nohpill Park. A new approach for time series forecasting based on genetic algorithm. In Frederick C. Harris Jr. and Fei Hu, editors, *Proceedings of the International Conference on Computer Applications in Industry and Engineering*, pages 226–231. ISCA, 2010.

[2] Neal Wagner, Zbigniew Michalewicz, Moutaz Khouja, and Rob Roy McGregor. Time series forecasting for dynamic environments: The dyfor genetic program model. *IEEE Transactions on Evolutionary Computation*, 11(4):433–452, August 2007.

[3] K. A. De Jong. *Analysis of Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 1975.

[4] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.

[5] Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel. A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, 1991.

[6] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, December 1997.

[7] Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[8] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth,

Australia, IEEE Service Center, Piscataway, NJ, 1995.

[9] Thomas Back, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, UK, 1st edition, 1997.

[10] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[11] Mengjie Zhang, Will Smart, Mengjie Zhang, Will Smart, Mengjie Zhang, and Will Smart. Multiclass object classification using genetic programming. In *In Applications of Evolutionary Computing, EvoWorkshops2004, volume 3005 of LNCS*, pages 369–378. Springer Verlag, 2004.

[12] Mengjie Zhang and Will Smart. Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification. *Pattern Recogn. Lett.*, 27(11):1266–1274, August 2006.

[13] Alaa F. Sheta, Peter Rausch, and Alaa S. Al-Afeef. A monitoring and control framework for lost foam casting manufacturing processes using genetic programming. *Int. J. Bio-Inspired Comput.*, 4(2):111–118, June 2012.

[14] Hossam Faris, Alaa Sheta, and Ertan Öznergiz. Modelling hot rolling manufacturing process using soft computing techniques. *International Journal of Computer Integrated Manufacturing*, 26(8):762–771, 2013.

[15] Hossam Faris and Alaa F Sheta. Identification of the tennessee eastman chemical process reactor using genetic programming. *International Journal of Advanced Science and Technology*, 50:121–140, 2013.

[16] Hossam Faris, Bashar Al-Shboul, and Nazeeh Ghatasheh. A genetic programming based framework for churn prediction in telecommunication industry. In *Computational Collective Intelligence. Technologies and Applications*, volume 8733 of *Lecture Notes in Computer Science*, pages 353–362. Springer International Publishing, 2014.

[17] Alaa F. Sheta, Hossam Faris, and Mouhammd Alkasassbeh. A genetic programming model for S&P 500 stock market prediction. *International Journal of Control and Automation*, 6(5):303–314, 2012.

[18] J. Koza. Evolving a computer program to generate random numbers using the genetic programming paradigm. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, La Jolla,CA, 1991.

[19] Abo El-Abbass Hussian, Alaa Sheta, Mahmoud Kamel, Mohamed Telbaney, and Ashraf Abdelwahab. Modeling of a winding machine using genetic programming. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 398–402, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 July 2000. IEEE Press.

[20] Dominic P. Searson, David E. Leahy, and Mark J. Willis. GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In *Proceedings of the International Multiconference of Engineers and Computer Scientists*, pages 77–80, Hong Kong, 17-19 March.

[21] Hui Qu and Xindan Li. Building technical trading system with genetic programming: A new method to test the efficiency of chinese stock markets. *Comput. Econ.*, 43(3):301–311, March 2014.

[22] M. A. Kaboudan. Genetic programming prediction of stock prices. *Comput. Econ.*, 16(3):207–236, December 2000.

[23] Mohammed E. El-Telbany. The egyptian stock market return prediction: A genetic programming approach. *Artificial Intelligence and Machine Learning (AIML)*, 5(3):7–12, 3 2005.

[24] Alaa Sheta and Hossam Faris. Improving production quality of a hot rolling industrial process via genetic programming model. *International Journal of Computer Applications in Technology*, 49(3/4), 2014. Special Issue on: "Computational Optimisation and Engineering Applications".

[25] Seyed Taghi Akhavan Niaki and Saeid Hoseinzade. Forecasting s&p 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*, 9(1):1–9, 2013.

[26] Sean Luke and Liviu Panait. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 829–836, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

Biographies

Alaa F. Sheta is a Professor with the Computers and Systems Department, Electronics Research Institute (ERI), Giza, Egypt. He received his PhD degree from the Computer Science Department, George Mason University, Fairfax, VA, USA in 1997. He received his B.E., M.Sc. degrees in Electronics and Communication Engineering from the Faculty of Engineering, Cairo University in 1988 and 1994, respectively. His main research area is in Evolutionary Computation, with a focus on Genetic Algorithms, Genetic Programming and applications. He is also interested in Particle Swarm Optimization, Differential Evolution, Cuckoo Search, etc. He has many publications in the area of image processing, software reliability modeling and software cost estimation. Alaa Sheta authored/co-authored about 100 papers in peer reviewed international journals, proceedings of the international conferences and book chapters. He is co-author of two books in the field of Landmine Detection, Classification and Image Reconstruction of Manufacturing Processes. He is the co-editor of the book: Business Intelligence and Performance Management - Theory, Systems and Industrial Applications by Springer Verlag, United Kingdom, published in March 2013.

Sara Elsir M. Ahmed received her B.Sc., M.Sc. degrees in Computer Science from the School of Mathematical Sciences, University of Khartoum in 1995 and 2003, respectively. Currently, Sara is a Ph.D. candidate with the Computer Science Department, College of Computer Science and Information Technology, Sudan University of Science and Technology, Sudan. Currently, Sara is a faculty member with Management Information System Department, Ahfad University for Woman, Sudan. Here research interests include Artificial Neural Networks, Genetic Algorithms, Genetic Programming, Data Mining, Modeling and Simulation of Nonlinear Systems.

Hossam Faris received his BA, M.Sc. degrees (with excellent rates) in Computer Science from Yarmouk University, Jordan and Al-Balqa Applied University, Jordan in 2004 and 2008, respectively. He is currently an Assistant professor with the Business Information Technology Department, King Abdullah II, School for Information Technology, University of Jordan. He has been awarded a full-time competition-based Ph.D. scholarship from the Italian Ministry of Education and Research to peruse his PhD degrees in e-Business at University of Salento, Italy, where he obtained his PhD degree in 2011. His research interests include: Applied Computational Intelligence, Evolutionary Computation, Knowledge Systems, Semantic Web and Ontologies.

www.icgst.com

A Model for Improving Classifier Accuracy using Outlier Analysis Methods

Lakshmi Sreenivasa Reddy. D, Ramchander. M
Associate Professor, Assistant Professor,Chaitanya Bharathi Institute of technology, Hyderabad, India
{drreddycsejntuh, ramchander}@cbit.ac.in,

Abstract

Outlier analysis is an important task for data mining to find outliers in datasets. Anomalies are objects; they have different behavior and do not follow with the remaining objects in the datasets. Outliers do not follow the rules formed by other data objects in the dataset. There are many methods available to detect outliers in numerical datasets. But limited methods are available for categorical datasets. New method has been proposed in this paper method to detect outliers in categorical data based on frequency of attribute value. In this paper we call these scores as BAD scores. This algorithm utilizes the frequency of each value in the dataset. It does not need any input to find number of outliers. Our algorithm shows better results in accuracy than AVF algorithm and Greedy. This proposed algorithm has almost reached to AVF in time complexity. This algorithm has been applied on Nursery dataset and Bank dataset taken from "UCI Machine Learning Repository". In this paper we have extended Normal distribution [11], and Fuzzy concept [12] to BAD score [13]. The experimental results show that it is an efficient in finding outliers from categorical dataset so that it improves the classifier accuracy.

Keywords: Data Mining, Outlier detection, BAD Score, NAVF, FuzzyAVF

1. Introduction

Outlier analysis is an important research field in many fields like networks, medical and Business. This analysis concentrates on data records with less frequency in the datasets. Most of the existing systems are use full for numerical attributes or ordinal attributes which can be converted to numeric and sometimes categorical attribute values can be converted into ordinal values there to numerical values. This process is not always preferable. This paper presented a novel model for finding anomalies in categorical data with the combination of BAD score, normal distribution and Fuzzy distribution. AVF method is the efficient method to detect outliers in categorical data both in time complexity and in accuracy. The mechanism in this AVF method is that, it calculates probability of each value in each data attribute and finds their average and selects top k-outliers based on the least AVF score. The parameter 'k' is used in this method. FPOF score can be found based on frequent patterns of each object which are adopted from Appriori algorithm [1]. This calculates frequent item sets from each object. From these frequencies it calculates FPOF score and finds the top k- outliers as the least k-FPOF scores. Time complexity is more for FPOF algorithm comparing with AVF algorithm. The parameters used in FPOF and FDOD are σ, which is a human decided threshold value to decide whether an item set in each data object is frequent or not and 'k', the number of outliers. The next method Greedy is based on Entropy score. For all these methods inputs are required. In our approach there is no need of inputs 'k' and 'σ'. Some of existing approaches are

2. Existing Approaches

2.1. Statistical Methods

Statistical Methods adopt a parametric model which describes the distribution of the records and the data was mostly univariate [3, 4]. Many drawbacks are there in statistical methods; these cannot find correct model for different datasets and the efficiency of these models decrease when the dimensions are increase [4]. To rectify this problem we can apply the Principle component method. Another method to handle high dimensional datasets is attribute relevance analysis .These ideas are not useful for more dimensions in any Dataset.

2.2. Distance Methods

These methods do not take any assumptions about the distribution of the data records because they should compute the distances between all records. But these methods make a high complexity. So these methods are impractical for large datasets with more records. Knorr's et al. [5], achieved some improvements in the distance-based algorithms, such as they have explained that apart of dataset records belong to each outlier must be less than some threshold value. Still it is an exponential on the number of nearest neighbors

2.3. Density Methods

Density base methods adopt on finding the density of the data records and identifying outliers as those lying in areas with low density. Breunig et al. have described a local outlier factor (LOF) to identify local outliers whether an object contains sufficient neighbor around it or not [6]. LOF decided a record as an outlier when the record LOF is less than the user defined threshold. Papadimitriou et al. described a similar method called Local Correlation Integral (LCI). This method selects the minimum points (min pts) in LOF through statistical methods in [7]. These density based methods have some advantages that they can detect outliers those are left by techniques with single, global criterion methods.

2.4. Deviation Methods

These methods find characteristics of objects instead of finding distances, densities and statistical parameters. The objects deviate from the given description is treated as outliers. The complexity is Linear with the dataset size. The terminology used in our paper is given below

TABLE 1. TERMINOLOGY

Term	Description
k	Target number of outliers
n	Number of objects in Dataset
m	Number of Attributes in Dataset
x_i	i^{th} object in Dataset ranging from 1 to n
A_j	j^{th} Attribute ranging from 1 to m
$D(A_j)$	Domain of distinct values of j^{th} attribute
x_{ij}	cell value in i^{th} object which takes from domain d_j of j^{th} attribute A_j
D	Dataset
V	Set of all distinct values in Dataset D
I	Item set
F	Frequent Item set
$f(x_{ij})$	Frequency of x_{ij} value
$FS(x_i)$	Set of frequent Item sets of x_i object
$IFS(x_i)$	Set of infrequent Item sets of x_i object
minsup	Minimum support of frequent item set
Support(I)	Support of Item set I
AVF	Attribute Value Frequency
FPOF	Frequent Pattern Outlier Factor
BAD	Boghapathi –Alisiri-Dirisinapu Factor

2.5. Greedy algorithm

If any dataset contain outliers then it deviates from its original behavior and this dataset gives us wrong results in data analysis. Greedy algorithm proposed the idea of finding a small subset of records; these contribute to eliminate the uncertainty of the dataset. This disturbance is also called entropy or disturbance. We can define it formally as 'let us take a dataset D with 'm' attributes A_1, A_2----- A_m and $d(A_j)$ is the domain of distinct values in the variable A_j, then the entropy of single attribute A_j is

$$E(Aj) = -\sum_{x \in D(A_j)} P(x)\log_2 P(x) \quad (1)$$

Since all attributes are independent to each other, Entropy of the entire dataset D= { A_1, A_2-------- A_m} is equal to the sum of the entropies of each one of the 'm' attributes, and it is defined as follows

$$E(A_1, A_2\text{--------} A_m) = E(A_1) + E(A_2) + \text{-----} E(A_m) \quad (2)$$

If we want to find entropy the Greedy algorithm takes k outliers as input [2]. All objects in the dataset are initially designated as non-outliers. Initially all attribute value's frequencies are computed and using these frequencies the initial entropy of the dataset is calculated. Then, Greedy algorithm scans k times over the data to determine the top k outliers keeping aside one non-outlier each time. While scanning each time every single non-outlier is temporarily removed from the dataset once and the total entropy is recalculated for the remaining dataset. For any non-outlier point that results in the maximum decrease for the entropy of the remaining dataset is the outlier data-point removed by the algorithm. The Greedy algorithm complexity is O(k *n*m*d), where k is the required number of outliers, n is the number of objects in the dataset D, m is the number of attributes in D, and d is the number of distinct attribute values, per attribute. Pseudo code for the Greedy Algorithm is as follows

Algorithm: Greedy

Input: Dataset – D
Target number of outliers – k
Output: k outliers detected
label all data points $x_1, x_2, \text{---} x_n$ as non-outliers
Calculate initial frequency of each attribute value and update hash table in each iteration.
calculate initial entropy
counter = 0
while (counter != k) do
    counter++
    while (not end of database) do
        read next record 'xi ' labeled non-outlier
        label 'xi' as outlier
        calculate decrease in entropy
        if (maximal decrease achieved by record 'o')
            update hash tables using 'o'
            add xi to set of outliers
        end if
    end while
end while

However entropy needs k as in put and need to find number of outliers more times to get optimal accuracy of any classification model.

2.6. Attribute Value Frequency (AVF) algorithm

The algorithm discussed above is linear with respect to data size and it needs k-scans each time. The other models also exist which are based on frequent item set mining (FIM) need to create a large space to store item sets, and then search for these sets in each and every data point


.These techniques can become very slow when we select low threshold value to find frequent item sets from dataset
Another simpler and faster approach to detect outliers that minimizes the scans over the data and does not need to create more space and more
Search for combinations of attribute values or item sets is Attribute Value Frequency (AVF) algorithm. An outlier point xi is defined based on the AVF Score below:

$$\text{AVF Score } (x_i) = \frac{1}{m}\sum_{j=1}^{m} f(x_{ij}) \qquad (3)$$

In this approach [1] again we need to find k-outliers many times to get optimal accuracy of any classification model. Pseudo code for the AVF Algorithm is as follows

Input: Database D (n points _ m attributes), Target number of outliers - k
Output: k detected outliers
Label all data points as non-outliers;
for each point xi, i = 1 to n do
for each attribute j, j = 1 to m do
Count frequency f(x_{ij})of attribute value x_{ij}; end end
for each point xi, i = 1 to n do
for each attribute j, j = 1 to m do
AVF Score(xi) += f(xij);
end
AVF Score(xi) /= m;
end
Return k outliers with mini (AVF Score)

The AVF algorithm time complexity is lesser comparing with Greedy algorithm .Since AVF needs only one scan to detect outliers, the time complexity is less. The complexity of AVF is O (n * m). AVF needs 'k' value as input to find 'k'-outliers.
In FPOF [8] this has discussed frequent pattern based outlier detection, in this too k-value and another parameter 'σ 'is required as threshold. This also discussed about frequent pattern based method to find infrequent object, in this too it requires k-value, and another parameter 'σ' as input. In the proposed model(Fuzzy AVF) it has been defined an optimal number of outliers in a single instance to get optimal precision in any classification model with good precision and low recall value. This method calculates 'k' value itself based on the frequency. Let us take the data set 'D' with 'm' attributes A1, A2----- Am and d (Ai) is the domain of distinct values in the variable Ai. k is the number of outliers which are normally distributed. To get 'k' this model used Gaussian theory (NAVF) and fuzzy theory. If the frequency is less than "mean-3 S.D" then this model uses fuzzy logic. This method uses AVF score formula, but no k-value is required

2.7. Frequent Pattern Outlier Factor FPOF algorithm

This algorithm utilizes the Appriori algorithm as first step to find all frequent Item sets .This method needs a human defined threshold value called" minimum support" as input to find frequent item sets. By taking this threshold value, it makes all combinations of values of each record and compares the frequency of each combination with threshold value and finds each combination whether it frequent or not. To find frequency of each combination, it needs one scan of the dataset. Even for one record it scans dataset so many times .If Dimensions are more with multiple values it is too cumbersome process, sometimes it is impossible. Even for ten dimensions with ten values each, it needs to generate one 10 million combinations approximately. These are the main disadvantages of the algorithms FPOF and FDOD. FPOF and FDOD algorithms take more memory and more time to generate combinations and their frequency.

2.8. BAD score Algorithm

The algorithms discussed above ,need many scans of dataset for each data object, but this algorithm needs only one scan of dataset for all records and it finds frequency of each value in dataset .This algorithm declares the records as outliers if any record value having frequency one . This algorithm finds the disturbance of each record in data set and finds k-records as those highest BAD scores [13]. This algorithm applied on Breast cancer data taken from UCI Machine Learning repository [10]; in this model we have defined the
1) Dataset as D= {A1, A2-------- Am},
2) D (Aj) = Domain of all distinct values in attribute 'j',
3) V=Set of all distinct values in dataset 'D'= D (A1)$\cup$ D (A2) $\cup$ D (A3)….D (Am) = {V1j, V2j V3j V4j ….Vkj}
Where $1 \le k \le n$ and $1 \le j \le m$ for each record, then our approach to find BAD score for each record as below

$$Score_1 = -\sum_{j=1}^{m}\left[\sum_{\forall Vkj \in D(Aj)\cap Xij \neq Vkj} \frac{(f(Vkj)-1)}{(n-1)} \log_{10}\left(\frac{(f(Vkj)-1)}{(n-1)}\right)\right] \qquad (4)$$

$$Score_2 = -\sum_{j=1}^{m}\left[\sum_{\forall Vkj \in D(Aj)\cap Xij \neq Vkj} \frac{(f(Vkj)}{(n-1)} \log_{10}\left(\frac{(f(Vkj)}{(n-1)}\right)\right] \qquad (5)$$

$$BAD_{Score} = \frac{1}{Score_1 + score_2} \qquad (6)$$

"BAD Score" Algorithm:
Input: Dataset D= {A1, A2-------- Am}, number of outliers-'k',
number of outliers-'k'
Output: k detected outliers
Step 1: Find frequencies of all values in dataset D and store them in V= { V1, V2, V3,…..Vk}
Step 2: For each record 'xi' in D
if f(X ij)=1 for any 'j'
go to step 6
else if Xij=Vkj and VkjεD(Aj)
find Score1
else if Xij≠Vkj and VkjεD(Aj)
find Score2
end else
end else
endif
end
Step -3: Find the Sum of Score1 and score2
Step 4: Then Find Reciprocal of the Sum
Step5: Find top k-Scores
Step6: return k-outliers

3. Our Approach (Normally Distributed BAD Score)

We have derived an approach based on mixing of normal distribution with BAD score(NBAD) like NAVF [11], and experiments conducted with this approach .This approach calculates N-seed values 'a' and 'b' as given below

$$b=\text{mean}\,(f(x_i)) \quad (7)$$

$$a=b-3*\text{std}\,(x_i),\ \text{if max}\,(f(x_i)) > 3*\text{std}\,(f(x_i)) \quad (8)$$

Comparing each record with these seed values this approach gives us outliers. This model conducted experiments on bank data taken from "UCI Machine Learning Repository" [10].Experimental results are given below

a) Our Approach2 (Fuzzy based BAD Score)

We have derived another approach based on fuzzy logic applied on BADscore (FBAD) like FAVF [12], and experiments conducted with this approach .This model calculates Fuzzy seed values 'a' 'b' and 'c' as given below

$$b = \text{mean}\,(fi) \quad (9)$$

$$a=\begin{cases} b-3*STD(fi) & if \quad \max(fi)>3*STD(fi) \\ b-2*STD(fi) & if \quad \max(fi)>2*STD(fi) \\ b-STD(fi) & if \quad otherwise \end{cases} \quad (10)$$

$$c=\begin{cases} b+3*STD(fi) & if \quad \max(fi)>3*STD(fi) \\ b+2*STD(fi) & if \quad \max(fi)>2*STD(fi) \\ b & if \quad otherwise \end{cases} \quad (11)$$

Here we derived the outliers based on fuzzy score based on S-fuzzy function .The S-fuzzy function is

$$S=\begin{cases} 0 & if \quad fi<a \\ 2\left\{\frac{fi-a}{c-a}\right\}^2 & if \quad a\le fi\le b \\ 1-2\left\{\frac{fi-a}{c-a}\right\}^2 & if \quad b\le fi\le c \\ 1 & if \quad fi>c \end{cases} \quad (12)$$

Where 'f_i' is the BAD score of' i^{th}' object in dataset 'D'. This method has been applied on bank data taken from "UCI Machine Learning Repository" [10].Experimental results are given below

4. Experimental results

In this paper this model has been used Bank from UCI Machine repository [10]. This method has implemented the approach of using MATLAB tool. We ran our experiments on a workstation with a Pentium(R) D, 2.80 GHz Processor and 1.24 GB of RAM. Bank data consists 45211 records and ten attributes "contact", "default", "education"," housing", "job", "loan", "marital status", "month", "poutcome" and a class label attribute 'Y'. This dataset contain two types of classes. one is yes ,other one is no, This data divided into two parts based on class attribute, first part contains 39922 records with "no" class, and second part contain 5299 records with "yes" label which are used as outliers in our experiment. We have done this separation by Clementine 11.1 tool. In first iteration 2645 sample records are selected randomly using Clementine tool; from each two records one is selected. These 2645 records are mixed up with part one which totals 42567 records and applied NBAD, NAVF, FBAD and FAVF to get outliers. Class attribute contains two values, all the remaining attributes "contact", "default", "education"," housing", "job", "loan", "marital status", "month", "poutcome" contain 3, 2, 4, 2, 12, 3,2, 12 and 4 values respectively, and the found outliers are given in Tables. Similarly 1058 records are selected randomly as one record from each five records and mixed up with first part and applied the same process .The results are given in the below Tables. Similarly one record is selected from each eight records and ten records and repeated the same process. Then we have applied our algorithms for on selected samples. Results are given in below Tables. This method has been implemented on Bank data which is taken from UCI Machine learning repository [10]. Comparison of results is given in Table II. Comparison graphs are given in the subsequent Figures.

TABLE 2. COMPARISON OF NUMBER OF OUTLIERS FOUND IN BANK DATA

Sample Method	NBAD	FBAD	NAVF	FAVF
1-in-2	933	363	274	279
1-in-5	543	264	366	199
1-in-8	396	231	152	154
1-in-10	336	187	126	126

From Table 2 results reveal that NBAD gives good number of outliers comparing with all other methods in any sample method. FBAD also gave good results comparing with NAVF and FAVF except at 1-in-5 sample. Graph is given in Figure 1

The same process has been applied on Nursery data with 6236 records [10]. The results show that NABD has found good number of outliers in this too. Graph is given in Figure 2

TABLE 3. COMPARISON OF NUMBER OF OUTLIERS FOUND IN NURSERY DATA

Sample Method	NBAD	FBAD	NAVF	FAVF
1-in-2	83	66	44	56
1-in-5	382	35	133	162
1-in-8	238	98	238	238
1-in-10	190	190	190	190

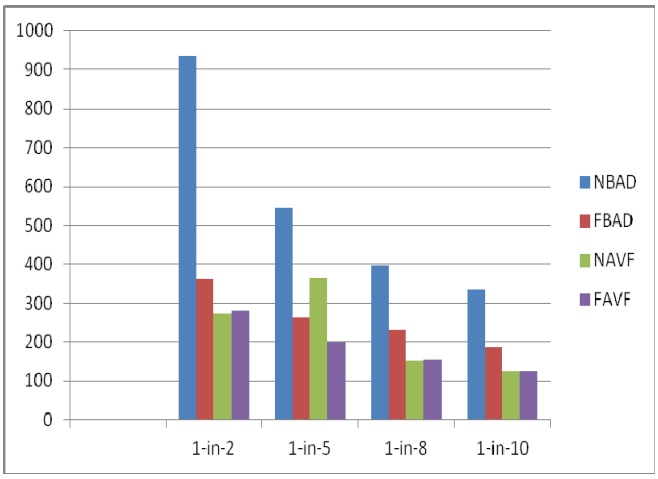



Figure 1 . Number of outliers found for Bank Data

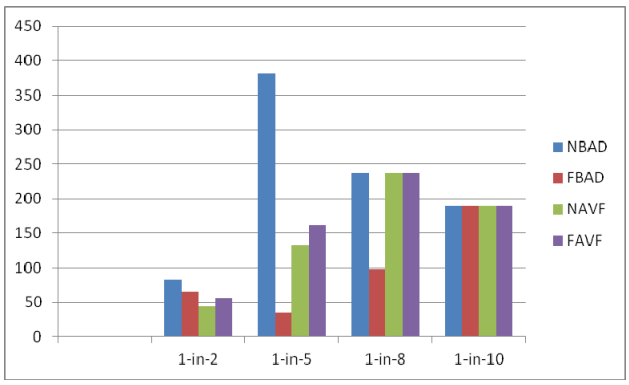


Figure 2 . Nursery Data

Different classifiers are tested on Bank data after deleting the outliers from NBAD, FBAD, NAVF, and FAVF.
Neural Network (NN), Logistic Regression (LR), CHAID, Decision Logic (DL) classifiers are applied by using Clementine 11.1 tool. The Accuracies of these models are given for the sample 1-in ten approaches in Table 4. Graph is given in Figure 3

TABLE 4. COMPARISON OF ACCURACIES OF CLASSIFIERS ON BANK DATA(1-IN-10 SAMPLE)

Classifier	NBAD	FBAD	NAVF	FAVF
NN	99.503	99.147	98.998	98.998
LR	99.503	99.147	98.998	98.998
CHAID	99.503	99.147	98.998	98.998
DL	93.732	80.906	37.202	37.2

From Table 4 all classifiers gave good results after deleting outliers by NBAD and FBAD has performed next .Accuracies graph is given in Figure 3.

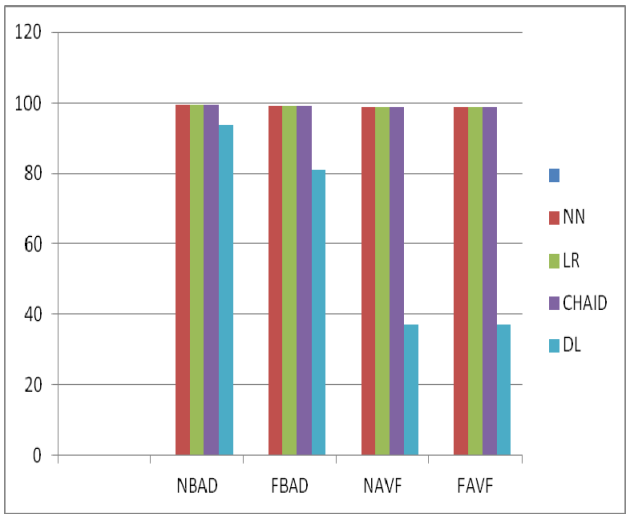


Figure 3 . Calssifiers accuracies for Bank Data

5. Conclusion and Future work

To sum up, this proposed method gives the more number of outliers comparing with existing models. Our model is good for categorical datasets to delete precise outliers. The combination of Normal distribution with our BAD score algorithm finds more outliers and train the classifiers with good accuracy. In future we can model different classifiers separately on mixed type of datasets.

6. References

[1] M. E. Otey, A. Ghoting, and and A. Parthasarathy, "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets," Data Mining and Knowledge Discovery

[2] He, Z., Deng, S., Xu, X., "A Fast Greedy algorithm for outlier mining", Proc. of PAKDD, 2006.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining: Pearson Addison-Wesley, 2005

[5] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," VLDB Journal, 2000.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density based local outliers," presented at ACM SIGMOD International Conference on Management of Data, 2000

[7] S. Papadimitriou, H. Kitawaga, P. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," presented at International Conference on Data Engineering, 2003

[8] Z. He, X. Xu, J. Huang, and S. Deng, "FP-Outlier: Frequent Pattern Based Outlier Detection", Computer Science and Information System (ComSIS'05)," 2005

[9] Shu Wu and Shengrui Wang, "Information-Theoretic Outlier Detection for Large-Scale Categorical Data, IEEE Transactions on Knowledge Engineering and Data Engineering,2011

[10] Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[11] LakshmiSreenivasaReddy.D, Dr.B.RaveendraBabu and Dr.A.Govardhan, "Outlier Analysis of Categorical Data using NAVF"', Informatica Economica vol 17, Cloud computing issue 1, 2013.

[12] LakshmiSreenivasaReddy.D, Dr.B.RaveendraBabu "Outlier Analysis of Categorical Data using FuzzyAVF", presented at IEEE international conference ICCPCT-2013,pp 1259-1263.

[13] LakshmiSreenivasaReddy.D, B.RaveendraBabu and etc, "Learning Styles Vs Suitable Courses" IEEE international conference -MITE-2013, pp 52-57.

[14] LakshmiSreenivasaReddy.D, B.RaveendraBabu "Efficient Model to Find Outliers in Categorical Data Using Outlier Factor by Infrequency", presented at IEEE international conference ICCPCT-2014, pp 1324-1328.

[15] M. E. Otey, A. Ghoting, and and A. Parthasarathy, "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets," Data Mining and Knowledge Discovery.

[16] He, Z., Deng, S., Xu, X., "A Fast Greedy algorithm for Outlier mining" Proc. of PAKDD, 2006.

[17] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[18] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining: Pearson Addison-Wesley, 2005

[19] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," VLDB Journal, 2000.

[20] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density based local outliers," presented at ACM SIGMOD International Conference on Management of Data, 2000

[21] S. Papadimitriou, H. Kitawaga, P. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," presented at International Conference on Data Engineering, 2003

[22] Z. He, X. Xu, J. Huang, and S. Deng, "FP-Outlier: Frequent Pattern Based Outlier Detection", Computer Science and Information System (ComSIS'05)," 2005

[23] Shu Wu and Shengrui Wang, "Information-Theoretic Outlier Detection for Large-Scale Categorical Data, IEEE Transactions on Knowledge Engineering and Data Engineering,2011

[24] Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[25] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases." Proceedings International Conference on Very Large Data Bases, 1994, pp. 487-499.

Biographies

Dr .Lakshmi Sreenivasareddy.D obtained his PhD and Masters Degree from JNTU University, Hyderabad. He worked as Director RISE Prakasam Group of Institutions, Ongole. Present he is working as Associate Professor in Chaitanya Bharathi Institute of Technology, Gandipet, Hyderabad. He has 12 years of teaching experience. His research area of interest is Data mining, Big data Analysis, Cloud Computing, Machine Learning.

M.Ramachander is working in Chaitanya Bharathi Institute of Technology. he has 10 years of teaching experience in CBIT, Gandipet, Hyderabad. He is perusing PhD in CSE from Osmania University. His research area of interest is Big data.

www.icgst.com

Expert System Development for the Fuzzy ANN Based Diagnosis of Brain Tumor

Nandita Pradhan[1], A.K. Sinha[2]
[1]*Department of ECE, United College of Engineering & Management,*
Naini, Allahabad, 211010, U.P., India.
[2]*Greater Noida Institute of Technology, Greater Noida,*
Greater Noida, 201 306, U.P., India.
E- Mail addresses: nanditapradhan123@yahoo.com, aksinha_1@yahoo.com

Abstract

A novel method for the development of an Expert System for the diagnosis of Brain Tumor has been presented in this paper. The knowledge has been acquired from clinical symptoms, neurological tests, cerebrospinal fluid (CSF) tests and feature vectors extracted from fluid attenuated inversion recovery (FLAIR) brain magnetic resonance (MR) images of the brain tumor patients from sizeable number of samples and data collected from different hospitals and nursing homes. Further each symptom case has nine most significant symptoms; each neurological examination case has seven most significant tests results; each CSF test has five parameter tests results and for MR brain image processing three empirically developed higher order wavelet and statistical functions are used and fuzzy C means algorithm is used for segmentation of intracranial image. Knowledge base thus acquired from symptoms, neurological tests, CSF tests results and from image processing are used for machine learning using fuzzy artificial neural network and expert system for brain tumor diagnosis is developed with an unique integrated approach. The results of proposed system are very promising and it is found that it can be effectively applied for automated tumor diagnosis. Final result depicted that 91.11% of correct diagnosis is obtained for brain tumor.

Keywords: *Brain Tumor, Clinical Symptoms and Neurological Tests, CSF Tests, Magnetic Resonance Images, Fuzzy Artificial Neural Network*

Nomenclature

MRI	Magnetic Resonance Images
FLAIR	Fluid Attenuated Magnetic Resonance
CSF	Cerebrospinal Fluid
NN	Neural Network
FANN	Fuzzy Artificial Neural Network
FBPA	Fuzzy back propagation Algorithm
MSE	Mean Square Error

1. Introduction

The prevalent practice of diagnosing brain tumor begins with noticing clinical symptoms followed by neurological tests, various pathological tests and brain MR image analysis. In order to arrive at definitive diagnosis, a team of experts from multiple disciplines like neurology, neuro-pathology, radiology etc. interpret and correlate the findings of various tests conducted on suspected brain tumor patients and conclusion is drawn. The entire process of making diagnosis is very complex and despite it consumes lots of time, money and human resources, robust and error free system could not be developed. To conquer this problem researchers are making continuous efforts to create a truly consistent expert system for the diagnosis of brain tumor which is a tough task.

In this paper a new approach for developing a comprehensive expert system with the help of machine learning using fuzzy artificial neural network (FANN) for diagnosing brain tumors is proposed. The schematic block diagram of developed expert system model is shown in figure 1. The knowledge-base acquired in this system is from brain magnetic resonance (MR) image analysis, clinical symptoms, results of neurological tests and cerebrospinal fluid (CSF) tests of brain tumor patients. In this work 9 categories of most prevalent symptoms, indicating for tumors are used. A neurological examination is usually first test given when a patient complains of symptoms that suggest a brain tumor. In this work patients are examined for 7 neurological tests. In many cases doctor also advise for CSF test through lumber puncture. Total 5 CSF parameter results from laboratory for different brain tumor patients are considered in this paper. Next step is advance brain imaging techniques, which provide meticulous anatomical delineation and are the principal tools for establishing that neurological symptoms are the consequences of a brain tumor.

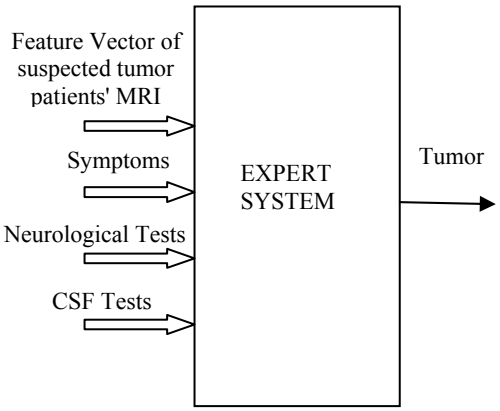


Figure 1. Schematic Diagram of Expert System

It is a challenging task to find exact transition from healthy tissues to edema to tumor, which is required for brain tumor management and treatment.

The automatic brain tumor and edema segmentation from magnetic resonance images (MRI) is a very important technique in medical imaging and is used in this research work. The findings of MRI based detection are established with the help of symptoms, neurological tests and CSF tests of brain tumor patients. In recent years the MR image segmentation techniques used by researchers are multi spectral analysis, fuzzy clustering techniques, classical pattern recognition methods, rule based system and through ANN (Pradhan and Sinha, 2008), level set evolution combining global smoothness (Dubey *et al*., 2009), with the help of knowledge based techniques, preprocessing and refinement on images are done to recognize and establish tissues in magnetic resonance images (Li *et al*., 1993) and (Li *et al*., 1996). Recently wavelets have found more and more applications in digital image processing. Many authors used properties of wavelet transform and multi- resolution theory for the segmentation of images (Taxt and Lundervold, 1994) and (Zhang and Desai, 2001).

At present in recent years utility of FLAIR images are found to be very useful and accepted widely in research area for the analysis and diagnosis of brain diseases. Researchers use FLAIR images to describe clinical applications. Tsuchiya *et al*. use FLAIR images for the study of intracranial tumors for more than 30 patients (Tsuchiya *et al*., 1996). The majority of the research work are focused on brain tumor segmentation only but the understanding of edema and its spread is very critical and decisive for the therapy planning, diagnosis, surgery and tumor management. Now in last one and half-decade research works have been done to detect both tumor and edema separately and simultaneously (Prastawa *et al*. 2004).

In this work, FLAIR images are used to segment tumor, edema and healthy tissues using empirically developed functions of higher order of Daubechies wavelet transform and statistical parameters as the elements of feature vector. The segmentation result using the newly developed composite feature vector is acceptable and MSE is very low (Pradhan and Sinha, 2009). Training of fuzzy artificial neural network using fuzzy back propagation algorithm is done for the detection of tumor and edema. The results thus obtained from fuzzy artificial neural network trained for MR brain images for tumor detection is augmented with the outputs of 3 other trained neural networks for symptoms, neurological tests and CSF tests to give final diagnosis of tumors.

Expert systems have a great potential in medical diagnosis. Systems (DENDRAL, 1965) and (MYCIN, 1972) were developed in Stanford University and were early Expert systems to assist physician to diagnose different diseases in 1970s. PUFF (Ajkins *et al*., 1983) is for diagnosing presence and severity of lung diseases and DeDombal's Leeds (De Dombal *et al*., 1972) is a abdominal pain system developed in the University of Leeds. Milord II has many modules like Terap-1A for pneumonia treatment and Ens A1 for diagnosing and orientation assistance in pedagogical process. Cadet developed in Tel Aviv, Israel is for early cancer finding. It is computerized clinical decision support system. CADIAG II (Vetterjein and Ciabattoni, 2010) is a successful expert system and has manifold application of the compositional rule of fuzzy inference based on symptoms, physical tests, pathological test and clinical tests and developed in 1980. Current version of Brain Tumor Diagnosis System (BTDS) is developed in Taiwan (Wang and Seng, 2007) to diagnose brain tumor. Examples from known data are used to induce rules that represent the knowledge in the domain. All these systems are not comprehensive and lacks in systemic approach.

Dxplain (Barnett *et al.* 1987) is a decision support system developed in 1987 at Massachusetts General Hospital, accepts sets of clinical findings to produce a ranked list of diagnosis. It has widespread use of for over 23 years and current version include 2400 diseases and over 5000 findings. XNEOR (Maria *et al.*, 1994) is an intricate rule based expert system developed in 1994 for treatment of pediatric brain tumors Medulloblastomas. It suggests various options of surgery, radiotherapy, chemotherapy or doses of medicines. ANDEXPERT (Bruning *et al*., 1997) developed in 1997 is knowledge based system for sonographic diagnosis of adnexal tumors and based on histopathological findings. ICD 10 based medical expert system (Chinniah and Muttan, 2009) developed in December 2009 provides advice, information and recommendation to the physician


using temporal logic. It provides fuzzy severity scale and weight factor for symptoms and disease. To detect suspected zone or tumors in medical images using hybrid systems is proposed (Benamrane *et al.*, 2005), which combines fuzzy neural network and expert systems. A systematic type II fuzzy expert system for diagnosing the human brain tumors using T_1 weighted MRI with contrast is presented (Zarandi *et al*. 2011) and found to be superior in recognizing the brain tumor and its grade than type I fuzzy expert systems. An expert system is developed for the diagnosis of thyroid diseases (Dogantekin *et al*., 2011) In this paper a generalized disciminant analysis and wavelet support vector machine system method for diagnosis of thyroid diseases in three phases is presented with classification accuracy of 91.86%.

All above-mentioned expert systems are developed for diagnosis or treatment of different types of diseases is fuzzy rule based, case based or hybrid systems. In recent years fuzzy set theory and fuzzy logic are applied successfully in medical expert systems. In our work a comprehensive expert system with the help of machine learning using FANNs which are trained using fuzzy back propagation algorithm (FBPA) is developed for the diagnosis of brain tumor which is minimal invasive. Brain tumor diagnosis is done without performing biopsy on the tumor tissues. For Expert system, knowledge has been acquired from clinical symptoms of 112 suspected tumor cases, neurological tests of 53 cases, CSF tests of 42 suspected patients and feature vectors extracted from FLAIR brain MR images of the 37 tumor patients. Brain MR images are processed and composite feature vectors, comprising of empirically developed higher order wavelet functions and statistical functions are extracted from the blocks of size 4×4 pixels of intra-cranial brain image. Finally for the definitive diagnosis, the findings of MR image analysis for the detection of brain tumors are established with the help of knowledge of clinical symptoms, Neurological tests and CSF tests.

Rest of the paper is organized as follows. Section 2 focuses on methodology and flow graph of the problem. Section 3 explains about image data and emphasizes on implementation part of the problem. Section 4 gives result and discussion and sections 5 and 6 tell about acknowledgement and references respectively.

2. Methodology and Algorithm

The objective of the research work is to develop an expert system in integrated framework to arrive at conclusive diagnosis of brain tumor using magnetic resonance images and supporting evidence on clinical symptoms, neurological tests and CSF tests. This Expert System is developed, broadly in two steps, stated as follows.

(i) Processing of brain MR images detecting pathological tissues.
(ii) Development of expert system by integrating clinical and neurological test results with the findings of MR image analysis as mentioned above in point (i).

The result of the first step obtained by feature vector extraction and segmentation of images is indicative of presence of unhealthy tissues, whereas to arrive at conclusive diagnosis second step is implemented. The development methodology for the Expert System for the diagnosis of brain tumor comprises of the development of knowledge base from clinical symptoms, neurological examinations, CSF tests results and analyzing brain MR images of tumor patients as mentioned below.

- Most prevalent symptoms, pointing towards tumor are categorized in nine categories and are treated as elements of feature vectors of FANN, NN-1 (figure 11). If output of NN-1, trained with fuzzy back propagation algorithm (FBPA) is high, it will suggest for brain tumor.
- Seven neurological examinations are selected and are treated as elements of feature vectors of FANN, NN-2 (figure 11). If output of NN-2, trained with FBPA is high then it will suggest for the presence of brain tumors.
- CSF tests are done for five parameters, which are treated as elements of feature vector and are used to train FANN, NN-3 (figure 11) with FBPA. If output is high it will suggest for the presence of tumor.
- Intracranial brain images of patient are processed for the segmentation of pathological tissues. For this FANN, NN-4 (figure 11) is trained with the help of FBPA to detect pathological tissues (Pradhan and Sinha, 2010). The flow diagram showing details of this work is shown in Figure 2.
- Fifth FANN, NN-5 (figure 11) is trained with the inputs from the outputs of above 4 fuzzy artificial neural networks (NN-1 to NN-4) and gives a definitive diagnosis for the presence of brain tumor.

Complete process can be overviewed through the detailed block diagram shown in figure 3.

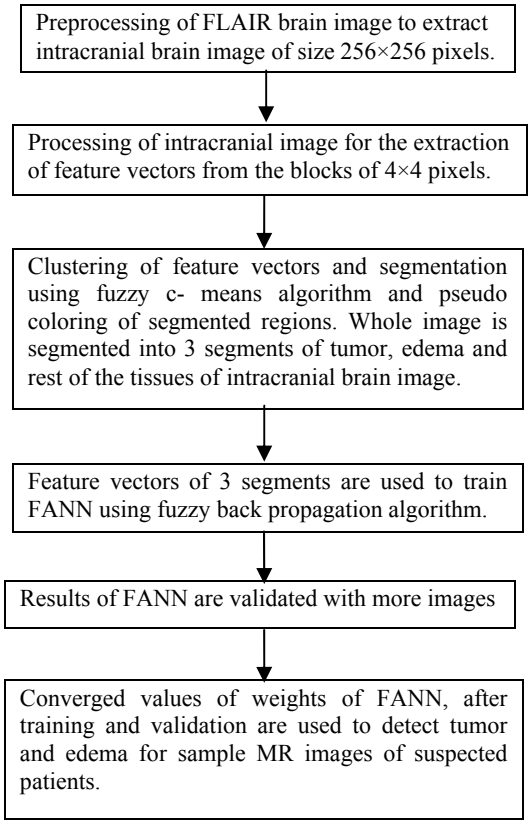



Figure 2. Diagram to show flow of brain image processing

2.1. Clinical Symptoms of Brain Tumor Patients

A brain tumor is defined as a growth of abnormal cells that is located in the brain itself. Brain tumor may be benign or malignant. Malignant tumor patients have very strong symptoms whereas benign tumor patients may have very weak symptoms or no symptoms at all. The brain has many different functions and there are different parts of brain that are responsible for these functions (NBTS, 2008). The effects of brain tumor depend entirely on the location, size and spread. Diagnosing a brain tumor starts with the symptoms of tumors. For this work most common symptoms of tumors are categorized in 9 categories as stated below.

- Mental changes and memory loss
- Headache
- Seizures
- Vomiting and Nausea
- Unsteadiness and problem with balancing and weakness in muscles
- Behavioral and cognitive problem
- Vision problems
- Hearing and Speech problem
- Depression, drowsiness and irritability

While these are the most common symptoms of brain tumor patients, they can also indicate other medical problems, so it is important to have definitive diagnosis of tumor. Symptoms of 112 suspected brain tumor patients collected from different hospitals and nursing homes are used in this research work.

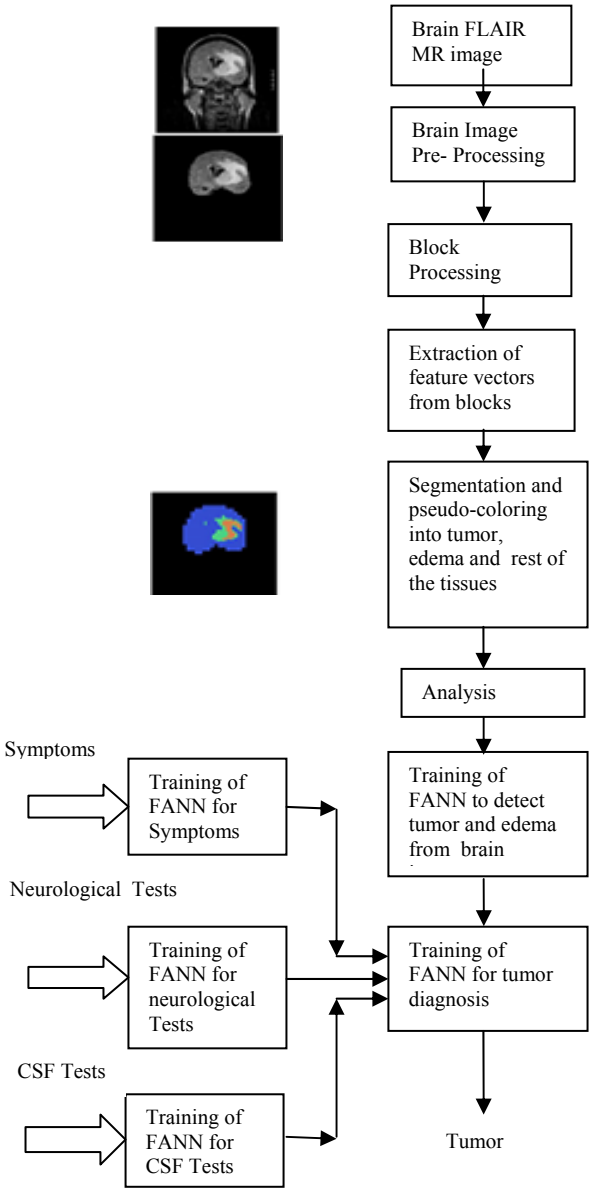


Figure 3. Complete Process Block Diagram

2.2 Neurological Tests Of Brain Tumor Patients

Observing the signs and symptoms of tumor, Neuro-Physician performs neurological tests on patient. A neurological examination is a series of tests to measure the function of patient's nervous system and physical and mental alertness. For this research paper these neurological tests of suspected tumor patient are categorized in 7 categories as stated below.

- Eye Reflex Test
- Hearing Test
- Reflex Test

- Balance/Co-ordination Test
- Touch/Muscle Test
- Head Movement Test
- Mental Status Test

If responses to the above tests are not normal, further tests or brain scans are required. 53 suspected brain tumor patients' neurological examinations results are available and used in this work.

2.3 CSF Tests of Brain Tumor Patients

CSF analysis is a set of laboratory tests that examine a sample of the fluid surrounding the brain and spinal cord for the presence of tumor. A lumber puncture (Spinal Tap) is used to obtain a sample of cerebrospinal fluid. Total of 5 parameters of CSF are used as 5 elements of feature vectors to train neural network. These parameters are;

- Opening Pressure
- Protein level
- White Blood Cell
- Red Blood Cells
- Glucose Level

Abnormalities in these tests may indicate towards many diseases including tumors. CSF test results are available for 42 suspected brain tumor patients as CSF test is not recommended for patients if intracranial pressure is high.

2.4 MRI of Brain for Tumor Patients

FLAIR MR images are very popular and useful for the diagnosis of brain diseases. These are used extensively for the analysis and diagnosis of brain tumor, acute hemorrhage, periventricular lesion, multiple sclerosis, head injury etc. With FLAIR MR images small and subtle wounds near the CSF become very clear and distinguishable in the back ground of CSF fluids because CSF is attenuated and looks dark and most of the injuries, tumors and edematous tissues appear bright (Pradhan and Sinha, 2009).

Researchers mainly focused on different techniques and methods for the segmentation of brain tumor only using MR image processing. However segmentation of tumor along with edema is required very much as the edema is another very important contributory factor in the symptoms of brain tumor. FLAIR images are proved better when compared to T2 and T1 weighted images as better discrimination between tumor and edema is obtained using FLAIR images.

Extracranial tissues are removed to make segmentation easy and for removing the possibilities of false segmentation. In this manuscript intracranial brain is extracted by removing eyes, skull tissues with the help of MATLAB image processing software. MR FLAIR brain images of 37 tumor patients are available in different planes for this research work.

For extraction of feature vectors, intracranial images are processed block wise. Whole image of size 256×256 is partitioned into blocks of 4×4 pixels and using MATLAB program feature vectors are extracted corresponding to each block. Feature vectors comprise of 3 elements, one element is higher order function related with pixel values of the block, and other two elements are higher order function of 2×2 wavelet coefficients of horizontal and diagonal frequency bands of blocks. The motivation for using the parameters extracted from high frequency bands is that they reflect texture properties and useful in segmentation process One level wavelet transform decomposes all blocks of size 4 × 4 pixels into four frequency bands of 2×2 coefficients. These four frequency bands are low frequency band (LB) and three high frequency bands (HB, VB, DB). LB gives approximate information and three high frequency bands give detail information in horizontal, vertical and diagonal directions.

In figure 4 (a) brain image and in figure 4(b) one level Daubechies wavelet transform of brain image is shown. In Fig. 4(b) the low frequency part is shown on the upper left corner, horizontal, vertical and diagonal sub images are displayed on upper right corner, lower left corner and lower right corner respectively. Here in this work, elements of feature vectors are obtained using coefficients of high frequency bands. Daubechies wavelet functions are having orthonormality and time frequency localization and making discrete wavelet analysis possible. Daubechies wavelet functions are optimized for tissue classification and gave better results.

The feature vector Fi for each block i∈I, where I is 2D brain image, is defined in equation (1) as

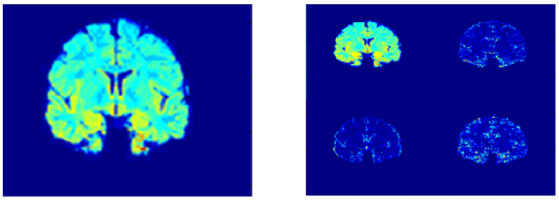
$$F_i\left[f_i^{(1)}, f_i^{(2)}, f_i^{(3)}\right] \quad (1)$$

Where $f_i^{(1)}$, $f_i^{(2)}$, $f_i^{(3)}$ are empirically developed elements of feature vector. First two elements are higher order functions of 2×2 wavelet coefficients of horizontal and diagonal frequency bands of blocks, which are empirical functions, developed by authors of this paper. After applying one level Daubechies transform, blocks of 4×4 pixels are decomposed into four sub images. In this work we have considered coefficients of horizontal band H1, H2, H3, H4 and coefficients of diagonal band D1, D2, D3, D4 to compute higher order wavelet features $f_i^{(1)}$ and $f_i^{(2)}$ as in equations (2) and (3) respectively.

$$f_i^{(1)} = \left\{{}^1\!/_4\ (H1^6 + H2^6 + H3^6 + H4^6)\right\}^{1/2} \quad (2)$$

$$f_i^{(2)} = \left\{{}^1\!/_4\ (D1^6 + D2^6 + D3^6 + D4^6)\right\}^{1/2} \quad (3)$$

Third element is a higher order empirical function and is related with pixel values of the block. Third element is defined using 16 pixel values P_1 to P_{16} of a block as given in equation (4).

(a) (b)

Figure 4 (a). Brain image (b) Brain image decomposes into four sub images after one level Daubechies WT

$$f_i^{(3)} = \{1/16\,(P_1^5 + P_2^5 + P_3^5 + \ldots + P_{16}^5)\} \quad (4)$$

In the current work brain images of size 256×256 pixels are taken. Image processing techniques are applied and 2D analysis is done for these images. One brain image is divided into 4096 blocks, each of size 4×4 =16 pixels and each block is having one feature vector. Each feature vector has 3 elements which describe characteristics of block. Total 4096 feature vectors clustered into three different clusters of tumor, edema and healthy tissues by fuzzy c-mean algorithm. Clustered feature vectors are used to form segmented image of original size 256×256.

Clustering algorithms are used to place similar patterns in the same cluster. The main difference between fuzzy clustering and other clustering techniques is that it generates fuzzy partitions of the data instead of hard partitions. In medical diagnosis systems, fuzzy c-means (FCM) algorithm gives the better results than hard- k means algorithm and is a very important supportive tool for the medical experts in diagnostic. In our paper clustering module is based on FCM algorithm (Jang *et al*., 2012) which is a fast algorithm. For FCM a program is developed in MATLAB that groups the feature vectors of image into 3 clusters.

3.Implementation

3.1 Structure of Artificial Neural Network trained with fuzzy back propagation algorithm

Fuzzy artificial neural networks (FANNs) have been taught to mimic the decision making process needed to perform the task of identification. Large MR data set having large number of independent variables and complex nonlinear relationship can be analyzed with the help of FANN. It is very useful tool in categorizing different brain tissues in terms of texture, intensity and contrast. Further separate FANNs are trained for clinical symptoms, neurological tests results and CSF tests results for indicating brain tumor. The program is made in MATLAB and algorithm used to train ANNs is the fuzzy back propagation algorithm (FBPA). Fuzzy back propagation network (FBPN) is hybrid architecture, which maps fuzzy inputs to crisp outputs. In this work FBPN is three-layered feed forward network having input layer, hidden layer and output layer. Both input and hidden layer are using fuzzy neurons. In this work in the fuzzy neuron, both input and weight vector are represented by triangular type of LR-type fuzzy numbers and can be represented as $(m, \alpha, \beta)_{LR}$. Here m is called mean value and α and β are left and right spreads respectively.

All FANNs (NN-1 to NN-5 of Figure11) used to implement expert system are using same training, transfer, learning and performance functions. All these networks are having three layered architecture as shown in figure 5 where n is number of input components of feature vector. NN-1 is for symptoms and using 9 components, 1 bias and 1 output; NN-2 is for neurological tests results and using 7 input components, 1 bias and 1 output; NN-3 is for CSF tests results and using 5 input components, 1 bias and 1 output; NN-4 is for FLAIR MR images and using 3 inputs, 1 bias and 1 output and finally NN-5 is using 4 inputs, 1 bias and 1 output.

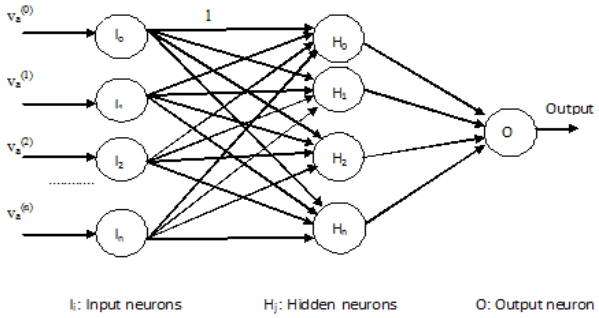


Figure 5 Architecture of Neural Network

Supervised learning algorithm is used for training of all fuzzy artificial neural networks which are using mean square error (MSE) function as the performance function. Learning of artificial neural network in this work is based on Gradient Descent Learning which is based on the minimization of error. The non linear transfer function used here is Sigmoidal transfer function which has greatest resemblance to biological neuron as compare to other transfer functions

3.2 Neurological, Pathological And Image Data

Knowledge base is prepared with symptoms of 87 patients (including 25 patients for which brain MR images are available for training), neurological tests of 41 patients (including 25 patients for which brain MR images are available for training), CSF tests of 30 patients (including 20 patients for which brain MR images are available for training) and feature vectors of brain MR image of 25 patients having both benign and malignant tumors. 12 other brain tumor patients' images and other data are available

using which system is evaluated. MR brain data is available from 1.5 Tesla SIEMENS MRI machine. For all patients several slices of T1, T1 contrast, T2, FLAIR images for axial, sagittal, coronal views are available. For this work FLAIR MR images in coronal plane are used with TI= 2300, TR=8000 and TE=120. In all acquisition, the field of view is 100mm×100mm and flip angle is 180°. Input data for fuzzy back propagation feed forward network are feature vectors containing three parameters, which are higher order functions developed by authors.

3.3 Training And Validation of FANNs NN-1, NN-2, NN-3 and NN-4

Fuzzy artificial neural network NN-1 (Fig.11) is trained and validated using fuzzy back propagation algorithm for symptoms of 87 suspected patients, collected from different hospitals and nursing homes. Total 9 symptoms mentioned in section II (A) are elements of feature vector F_s which is described in equation (5). Sample feature vectors for symptoms are given in Table I.

$$F_s = [f_s^{(1)}, f_s^{(2)}, f_s^{(3)}, f_s^{(4)}, f_s^{(5)}, f_s^{(6)}, f_s^{(7)}, f_s^{(8)}, f_s^{(9)}] \quad (5)$$

Where $f_s^{(1)}$ to $f_s^{(9)}$ are symptom elements of F_s may be strong, moderate, weak or zero. Depending on different combinations of elements, if NN-1 gives high output it will indicate for tumors. Sample feature vectors for symptoms are given in Table 1. The performance measure mean square error (MSE) is derived which is as low as 7.83206e-019 as shown in figure 6.

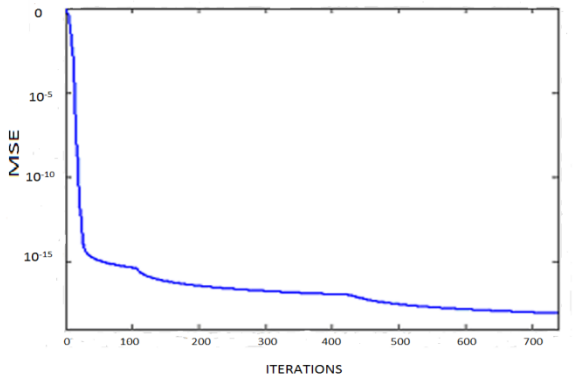


Figure 6. Performance measure for fuzzy NN-1 trained for symptoms of tumors

Fuzzy artificial neural network NN-2 (Fig.11) is trained and validated using fuzzy back propagation algorithm for neurological tests of 41 suspected brain tumor patients. Seven neurological examinations mentioned in section II (B) are elements of feature vector F_n which is described below in equation (6).

$$F_n = [f_n^{(1)}, f_n^{(2)}, f_n^{(3)}, f_n^{(4)}, f_n^{(5)}, f_n^{(6)}, f_n^{(7)}] \quad (6)$$

Where $f_n^{(1)}$ to $f_n^{(7)}$ are elements of F_n. If output of NN-2 is high then neurological tests may suggest for tumor. Sample feature vectors for neurological tests are given in Table 2. The performance measure MSE of the NN-2 is derived which is as low as 3.21698e-016 as shown in figure 7.

Fuzzy artificial neural network NN-3 (figure 11) is trained using fuzzy back propagation algorithm for CSF tests of 30 suspected tumor patients. Five CSF parameters mentioned in section II (C) are elements of feature vector F_c which is described in equation (7).

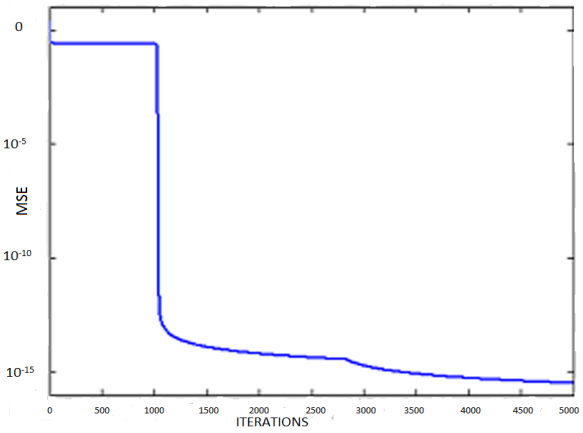


Figure 7. Performance measure for fuzzy NN-2 trained for neurological tests of tumor patients

$$F_c = [f_c^{(1)}, f_c^{(2)}, f_c^{(3)}, f_c^{(4)}, f_c^{(5)}] \quad (7)$$

Where $f_c^{(1)}$ to $f_c^{(5)}$ are elements of F_c. Sample feature vectors for CSF tests are shown in Table 3. If FANN output is high it may indicate towards tumor. The performance measure of the NN-3 is as low as 3.50452e-019 as shown in figure 8.

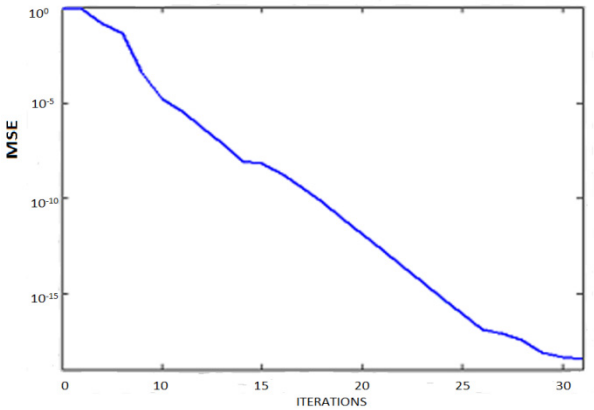


Figure 8. Performance measure for fuzzy NN-3 trained for CSF parameters of tumor patients

Fuzzy artificial neural network NN-4 (figure 11) is trained using fuzzy back propagation algorithm for feature vectors of tumor segments of 25 suspected tumor patients. Three empirically developed higher order wavelet and statistical components as mentioned in equations (2 to 4) of section II (D) are elements of feature vector F_i of (1). Sample feature vectors of suspected tumor segments of brain MR

images are shown in Table 4. If FANN output is high it may indicate towards tumor. NN-4 is used to identify tumor tissues in the images of 12 unknown suspected tumor patients. Network is tested with whole image and detected pathological tissues in 12 different tumor patient cases and verified by radiologists. The results in this work are very encouraging and computational speed and efficiency is satisfactory. The performance measure of the NN-4 is as low as 4.363e-017 as shown in figure 9.

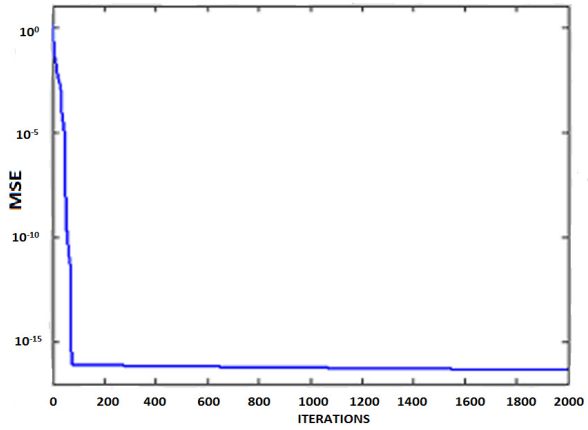


Figure 9. Performance measure for fuzzy NN-4 trained to detect pathological tissues

For segmenting the image the software developed in this work in MATLAB, clusters 64×64=4096 input feature vectors of image into different clusters of tumor, edema and healthy tissues. Clustered feature vectors are used to form segmented image of original size 256×256 (Pradhan and Sinha, 2010). Segmentation result after pseudo coloring is shown in Fig. 10 for four patients with tumor and edema. In figure 10 (a1 to a4) original intracranial images and in Fig. 10 (b1 to b4) block wise reconstructed segmented image with 3 segments showing tumor (red), edema (blue), healthy tissues (green) and background (black) are shown. In figure 10 (c1 to c4) segmented tumor and edema tissues are only shown.

Validation of automatic method for brain image segmentation is also done and performance of segmentation is accessed quantitatively through a simple method developed. For unavailability of standard data on tumor affected brain MRI, the automatic segmentation results are compared with manually segmented images generated with the help of experts. A flat tolerance of 10% is taken considering all possible errors into account which mainly include error due to block processing used for automatic segmentation process and errors in segmentation of manually segmented brain images due to human errors. Comparative quantitative analysis shows total segmentation error is in acceptable limits and satisfactory.

3.4 Complete Expert System

The complete system developed as shown in figure 11 to diagnose tumor includes four FANN, NN-1 to NN-4, trained with fuzzy back propagation algorithm are for symptoms, neurological tests, CSF tests and for intra-cranial MR images to detect tumor . Outputs of NN-1 to NN-4 are inputs of FANN NN-5, which is also trained using fuzzy back propagation for definitive diagnosis of tumor.

4. Result and Conclusions

A comprehensive expert system is implemented with the data of substantial number of patients. The result of proposed approach is very promising and efficiently applied for tumor diagnosis. Trained and validated system is tested with the data of 12 unknown brain tumor patients with 90 samples out of which in 82 cases tumor tissue diagnosis is matching with the opinion of radiologists and neurologists. Performance measure of the fuzzy artificial neural network, NN-5 is 1.486e-017 as shown in figure 12.
Proposed Expert system is implemented in two stages. In first stage 4 individual neural networks are trained and validated for symptoms, neurological examinations, CSF tests and feature vectors extracted from images. In second stage outputs of all 4 neural networks NN1-NN4 are given to the inputs of NN-5 and then output of NN- 5 is taken as final diagnosis of tumor.

Neural network NN-1 is tested for 25 patients for symptoms to be present and indicating towards tumor. Correct diagnosis is obtained for 23 tumor patients giving 92% of correct result. Similarly NN-2 is tested for 12 patients for neurological examinations indicating towards tumor. In one case result mismatch occurred as compared to the neurologists' diagnosis.
Similarly for CSF tests NN-3 is tested for 12 patients and gave 1 incorrect diagnosis when compared to the pathologist's opinion. NN-4 is tested with 90 samples taken from 12 patients. It gives correct diagnosis for 85 samples. Outputs of all 4 neural networks are given to the inputs of NN-5 and tested for 90 samples of 12 patients from which 91.11% of correct diagnosis is obtained Block wise segmentation of MR images improves the computational speed of the segmentation process.

Table 1
SAMPLE FEATURE VECTORS OF PATIENTS FOR SYMPTOMS

S. No.	$f_s^{(1)}$	$f_s^{(2)}$	$f_s^{(3)}$	$f_s^{(4)}$	$f_s^{(5)}$	$f_s^{(6)}$	$f_s^{(7)}$	$f_s^{(8)}$	$f_s^{(9)}$	Suggesting Brain Tumor
Patient 1	1	1	0	0	0	0	0	0	1	Yes
Patient 2	0	0.2	1	1	0	0.5	0	0	0.5	Yes
Patient 3	0	0	0.2	0	0	0	0	0	0	No
Patient 4	1	1	0	1	0.2	1	0	0.5	0	Yes
Patient 5	0	0	0	0.2	0	0.2	0.2	0.2	0	No
Patient 6	0.5	0.5	1	1	0	1	1	1	0.5	Yes
Patient 7	0	0.5	0.5	0.5	0	0	1	0	0	Yes
Patient 8	0	0	0	0.5	0	0	0.2	0.2	0	No

Where 0 indicates no symptoms, 0.2 weak symptoms, 0.5 moderate symptoms and 1 indicates strong symptoms

Table 2
SAMPLE FEATURE VECTORS OF PATIENTS FOR NEUROLOGICAL EXAMINATIONS

S. No.	$f_n^{(1)}$	$f_n^{(2)}$	$f_n^{(3)}$	$f_n^{(4)}$	$f_n^{(5)}$	$f_n^{(6)}$	$f_n^{(7)}$	Suggesting Brain Tumor
Patient 1	1	1	1	1	1	1	1	Yes
Patient 2	0	1	1	1	1	1	1	Yes
Patient 3	0	1	0	0	0	0	0	No
Patient 4	0	0	0	1	0	0	0	No
Patient 5	1	1	0	0	0	0	1	Yes
Patient 6	1	1	1	1	0	0	0	Yes

Where 1 indicates that neurological test is positive and 0 indicates that neurological test is negative

Table 3
SAMPLE FEATURE VECTORS OF PATIENTS FOR CSF

S. No.	$f_c^{(1)}$	$f_c^{(2)}$	$f_c^{(3)}$	$f_c^{(4)}$	$f_c^{(5)}$	Suggesting Brain Tumor
Patient 1	263	117	202	410	23	Yes
Patient 2	288	126	232	285	35	Yes
Patient 3	135	33	5	0	63	No
Patient 4	163	23	14	2	51	No
Patient 5	302	136	220	402	32	Yes
Patient 6	110	33	05	2	72	No

Table 4
SAMPLE FEATURE VECTORS OF SUSPECTED TUMOR SEGMENTS OF BRAIN MR IMAGES

FV No.	Horizontal Wavelet Function	Diagonal Wavelet Function	Higher Order Statistical Function	Suggesting Brain Tumor
1.	0.0065	0.005	0.0077	Yes
2.	0.1277	0.0001	0.1005	No
3.	0.0091	0.0013	0.0182	Yes
4.	0.0339	0.0037	0.1215	Yes
5.	0.0082	0.0044	0.0706	No
6.	0.0907	0.0003	0.0708	No
7.	0.0085	0.0002	0.0745	Yes
8.	0.0023	0.0003	0.0026	No
9.	0.0047	0.0002	0.0946	Yes


BPA based ANN offers promising results in the task of classifying brain tissues. Results of neural networks after comparing with neurologists and radiologists findings are shown below in Table 5.In this paper an expert system is developed which can be efficiently applied for brain tumor diagnosis. Performance of system is evaluated and found that for 91.11% samples correct diagnosis is done. The Expert system developed here can be used for the diagnosis of brain tumor and can be very helpful to the neurologists and radiologists for the definitive diagnosis of tumor.

5. Acknowledgement

The authors acknowledge all the help and support provided by senior radiologists Dr. Ajay Sharma, Dr. Yashpal Dahia and Dr. Manpreet of Aashlok Hospital and Delhi MR and CT Scan Centre, New Delhi, India. Authors also acknowledge the help and data provided by Kailash Hospital, Noida, India to make this research possible.

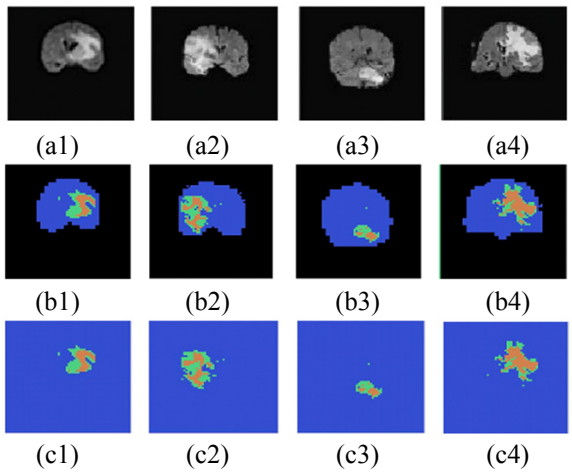


Figure 10 (a1 to a4) Intracranial Brain (b1 to b4) Segmented brain image (c1 to c4) Segmented tumor and edema

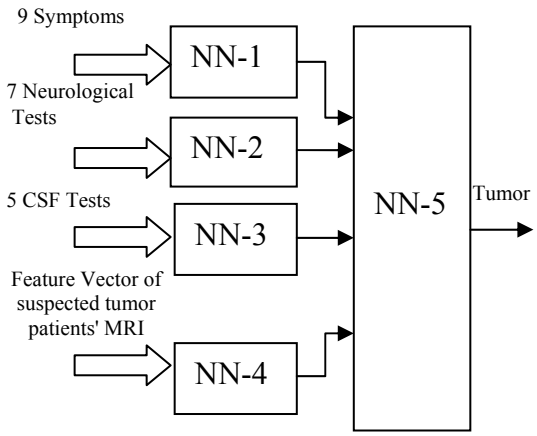


Figure 11. Complete Fuzzy ANN Expert System

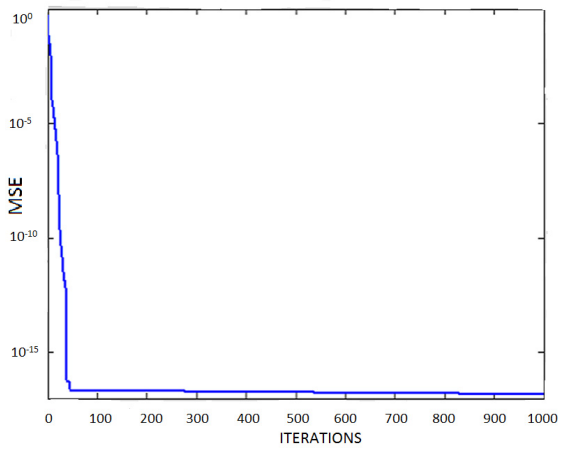


Figure 12 Performance measure for fuzzy NN- 5 trained to give definitive diagnosis of brain tumor

Table 5
RESULTS OF FUZZY ANN

S. No.	Neural Network	No. of patients (No. of Samples)	No. of Correct Diagnosis	% Correct Result
1	NN-1	25	23	%92
2	NN-2	12	11	%91.66
3	NN-3	12	11	%91.66
4	NN-4	90	85	%94.4
5	NN-5	90	82	%91.11

6. References

[1] J.S. Ajkins , J. C. Kunz, E. H. Shrtliffe, and R. J. Fallat "PUFF: An expert system for interpretation of pulmonary function data", *Comput Biomed Res* Vol.16, Issue.3, pp.199-200, 1983.

[2] G. O. Barnet, J. J. Cimino, J. A. Hupp, and E. P. Hoffer "DXplain. An evolving diagnostic decision-support system". *JAMA*. vol.258, no.1, pp. 67-74, 1987.

[3] N. Benamran, A. Freville and R. Nekkache, A Hybrid Fuzzy Neural Networks for the Detection of Tumors in Medical Images, *Americal Journal of Applied Sciences*, ISSN 1546-9239, Vol.2, Issue.4, pp.892-896, 2005.

[4] J. Bruning, Becker, R., Entezami, M., Loy, V., Vonk, R., Weitzel , H., and Tolxdorff, T. , Knowledge-Based System ADNEXPERT to Assist

the Sonographic Diagnosis of Adnexal Tumors, *Methods of Information in Medicine*, Vol.36, Issue.3, pp.201-206, 1997.

[5] P. Chinniah and S. Muttan "ICD 10 based Medical Expert System" in *International Journal of Computer Science and Information Security,* vol.6,no.3 , pp-084-089, 2009.

[6] F.T. De Dombal, D. J. Leaper, J. R. Staniland, A. P. McCann, and J. C. Horrocks "Computer-aided diagnosis of acute abdominal pain", *British Medical Journal*, Vol.2, no. 5804, pp.9-13, 1972.

[7] DENDRAL, Stanford University , available at http://www.comp.dit.ie/rlawlor/KE/notes/Dendral.pdf, 1965

[8] E. Dognaketin, Akif Dognaketin, and Derya Avic "An Expert System based on Generalized Disciminant Analysis and wavelet support vector machine for diagnosis of thyroid diseases" *Expert System with applications*, vol. 38, no. 1, pp 146-150, 2011.

[9] R.B. Dubey, Hanmandalu, M. And S. K. Gupta "Semi Automatic Segmentation of MRI Brain Tumor" ICGST GVIP, vol. no. 9, no. 4,pp 33-40, 2009.

[10] J.S.R. Jang, C. T. Sun, and E. Mizutani, "Data Clustering Algorithm" in *Neuro-Fuzzy AND Soft Computing- A Computational Approach to Learning and Machine Intelligence*, ISBN-978-81-203-2243-1, PHI, pp 425-427, 2012.

[11] C. Li, D. Goldgof, and L. Hall, "Automatic segmentation and tissue labeling of MR brain images," *IEEE TMI* vol. 12, pp 740- 750, 1993.

[12] X. Li , S. Bhide, and M. Kabuka " Labeling of MR brain images using Boolean neural network" IEEE TMI vol.15, no. 2, pp 628-638, 1996.

[13] B. L. Maria, F.A. Lamba, D. Dankel II, S. Chakravarthy, S. Tufekci, R. Marcus and A. Kedar, XNEOr : Development and Evaluation of an Expert System to Improve the Quality and Cost of Decision –Making in Neuro-Oncology" *Proc Annu Symp Comput Appl. Med Care*, pp.678-683, 1994.

[14] MYCIN, Stanford University, available at http://neamh.cns.uni.edu/MedInfo/mycin.html, 1972

[15] NBTS, available at http://www.braintumor.org, 2008

[16] N. Pradhan and A.K. Sinha, "Computer- Aided Detection of Tumor and Edema in Brain FLAIR Magnetic Resonance Image Using ANN", Proceedings of *International Conference on Computer- Aided Diagnosis of SPIE Symposium on Medical Imaging,* San Diego, California, USA, 2008, Vol. 6915, pp. 6915 1W-1 to 1W-9, 2008.

[17] N. Pradhan and A.K. Sinha, "Development of a composite feature vector for the detection of pathological and healthy tissues in FLAIR MR images of brain" ICGST *Bioinformatics and Medical Engineering Journal*, vol.10, pp 1-11, 2009.

[18] N. Pradhan and A. K. Sinha, "Intelligent computing for the analysis of Brain Magnetic Resonance Images", Proceedings of *IEEE Conference on Integrated Intelligent Computing, ICIIC 2010*, Bangalore, India, PP 211-217, 2010.

[19] M. Prastawa, E. Bullitt, S. Ho, and G. Gerig "A brain tumor segmentation framework based on outlier detection" *Elsevier's Medical Image Analysis* 8, pp 275-283, 2004.

[20] T. Taxt, and A. Lundervold, "Multispectral analysis of the brain using magnetic resonance imaging" *IEEE TMI* vol. 13, pp 470-481, 1994.

[21] K. Tsuchya, Y. Mizutani, and J. Hachiya "Preliminary evaluation of fluid attenuated inversion recovery MR in the diagnosis of intracranial tumors" AJNR Neuroradiol , vol. 17, no. 6, pp 1081-1086, 1996.

[22] T. Vetteriein, and A. Ciabattoni "CADIAG II", *Journal on Fuzzy sets and systems,* vol.161, no. 14., 2010

[23] Wang, C. H. and S. S. Tseng "A brain tumor detection system with automatic learning abilities" III annual , IEEE symposium on Computer based medical systems, Chapel Hill, NC, pp.313-320, 1990.

[24] M. H. F. Zarandi, M. Zarinbal and M. Izadi, Systmatic image processing for diagnosing brain tumors : A Type-II fuzzy expert system approach, Journal Science Direct Applied Soft Computing, Vol.11, Issue 1, pp.285-294, 2011.

[25] X. P. Zhang and M. D. Desai "Segmentation of Bright Targets Using wavelets and Adaptive Thresholding" in IEEE transaction on Image Processing, vol.- 10 ,no. 7, pp 1020-1030, 2001.

Biographies

Nandita Pradhan is Professor and Head of Department in, UCEM, Allahabad in U.P. Technical University, Lucknow, India. She received her Bachelor of Engineering in 1985 and Master of Engineering in 1991, both with distinction from Government Engineering College, Jabalpur, India. She received her Ph.D. degree from Uttar Pradesh Technical University, Lucknow, India in 2014. She has published many papers in the Conferences/ journals. She is MIEEE, life member of ISTE, India and IETE, India. Her areas of interest are Image Processing, Medical Imaging, Biomedical Instrumentation and Digital Signal Processing. Her research topic is "Expert System Development for Biomedical Studies using MRI". She has published papers in this area in many journals and conferences e.g. IC on Computer Aided Diagnosis in SPIE symposium on Medical Imaging in San Diego, IEEE conference on Integrated Intelligent Computing, ICIIC in Bangalore, in ICGST Bioinformatics and Medical Engineering journal, IJ of Computer Science and Information Technology, IJ of Information Technology and Knowledge Management etc.

Ashok Kumar Sinha is a Professor and Director in GNNIT Engineering College, Greater Noida, India. His area of interest are Signal and Image Processing analysis, Artificial Intelligence, Neural Networks and Systems Modeling etc. He did his B. Tech, from Bihar University in 1966, M. Tech. from IIT, Delhi in 1976 and Ph.D. from IIT, Delhi in 1981. He is IEEE affiliate of Computer Society and life member of Indian Society of Technical Education. He has been consultant to Planning Commission, Government of India and worked on several projects on systems analysis and techno-economic feasibility studies. He has publications in the area of transportation system modeling, satellite image processing and MRI image processing in national and international journals and conference proceedings. He has chaired technical sessions in various conferences in India.

www.icgst.com

HPC based Modeling, Analyzing and Forecasting of a Century of Climate Big Data

K. ElDahshan and H. Mancy
Dept. of mathematics, Computer science Division, Faculty of science, Al-Azhar University, Cairo, Egypt
[dahshan, dr.hendfathi]@azhar.edu.eg,
http://www.azhar.edu.eg

Abstract

This article focuses mainly on climate data forecasting. To achieve this with a reasonable accuracy we rely on deep processing of big data using High Performance Computing (HPC). To accomplish this target, 15 years of climate data are employed to generate a huge amount of forecasting information. In general, handling of big data should begin with modeling and analysis. This research work shows how to build a model to analyze big data. Climate Forecasting is done using big data technologies and tools such as; WRF and HPC.

Keywords:
Big Data, Big Data Technologies, Weather Forecasting, Big Data Modeling, Data Analysis, HPC, WRF, Climate Data.

Nomenclatures

IDC	International Data Corporation
HDFS	Hadoop Distributed File System
WRF	Weather Research and Forecasting Model
NCEP	National Centers for Environmental Prediction
GDAS	Global Data Assimilation System
GEFS	Global Ensemble Forecast System
GFS	Global Forecast System
NAM	North American Mesoscale
RAP	Rapid Refresh
RUC	Rapid Update Cycle
NWP	Numerical Weather Prediction
HPC	High Performance computing
NOAA	National Oceanic and Atmospheric Administration
BA	Bibliotheca Alexandrina
FLOPS	Floating-point Operations Per Second

1 Introduction

Nowadays, Egypt is considered to be one of the countries exposed to climate change. Climate change is expected to affect all development sectors in Egypt including agriculture, tourism, public health and human life in general. Data scientists start studying and analyzing climate data in Egypt to predict environmental impacts of climate change. Climate data is considered a big data example. Big data characteristics are discussed in section (2).
Big data technologies are being used to develop a model for climate changes and explaining their environmental impacts. Weather forecasting depends on computer based. This models handle different atmospheric factors [9].
Big data gathered, processed and used for weather forecasting results in wide computational operations. More than fifteen percent of the list of top 500 supercomputers, that has specified application areas, is dedicated merely to weather and climate research [5].
Data analysis is the most important and the final phase that explains the value of big data. In this phase we can extract different levels of useful information, decisions or suggestions for various fields [20].
In this paper we study big data; big data management phases, big data technologies and its tools that help us to create a model used for analyzing and forecasting climate big data.

The remainder of the paper is organized as follows: Section (2) focuses on the concepts of Big Data Section (3) focuses on some important phases of big data; specifically big data analysis Section (4) presents a comparative study of the main big data processing methods and techniques Section (5) details information about HPC and BA-HPC Section (6) explains weather forecasting models and focuses on

WRF Section (7) summaries the related work Section (8) presents a new model to handle different phases of managing big data using big data technologies and its tools Section (9) shows how to apply the new model on a case study.

2 Big Data

In this section we introduce the concepts of big data.

2.1 Big Data Definition

Big data is identified as a huge amount of data collected through time and is difficult to handle using traditional database management tools. Social media activities, business activities, photos, videos, sensors, e-mails, text files and application logs are sources of big data [25].
Big data may be described by velocity, volume and variety characteristics. To extract useful information from big data, one needs a great processing power, analytical capabilities and skills [3].
Big data generally ranges from several TB (Terabytes) to several PB (Petabytes) and even EB (Exabytes) [17].
The National Institute of Standards and Technology NIST considers "big data as data which exceed(s) the capacity or capabilities of the current or conventional systems. In other words, the notion of "big" is relative to the current standard of computation" [4].

2.2 Big Data Characteristics

Gartner analyst Doug Laney describes big data as having three dimensions: volume, variety, and velocity. Thus, IDC (International Data Corporation) defined it as follows: "Big data technologies describe a new generation of technologies and architectures designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis [11]". Other characteristics may include: veracity, validity, volatility and value.

2.3 Big Data Types

There are three Types of big data; structured, unstructured and semi-structured data [24].
Table 1 shows a comparative study of big data types.

2.4 Big Data Technologies and Tools

Many technologies and tools are used to store, manage and analyze big data. Technologies and tools that support big data techniques include, but are not limited to [16]:

- Hadoop is an open source software framework for processing large amounts of data on a distributed system for a given problem. It is designed to store, manage, and analyze hundreds of Terabytes and even petabytes of data. It is considered as one of the big data analysis platforms supporting Windows, Linux, and OS X operating systems. It is easy to work with multiple data sources, large scale processing and large volumes of data, such as transaction data, social media data and weather location-based data. Hadoop components include:
  - **Hadoop Distributed File System (HDFS)** which is a java-based file system that provides scalable, credible and redundant data storage designed to cover large clusters of commodity hardware.
  - **MapReduce** engine which is a framework for distributed processing of large data sets on computer clusters. It handles scheduling tasks, monitoring them and re-executing any failed tasks.
- NoSQL database (Not Only SQL) is a new generation of databases with new characteristics; being non-relational, distributed, horizontally scalable, schema-free, supports easy replication, supports simple API and keeps large amounts of data. There are several database types that fit into this generation; such as key-value stores and document stores. This database types focus on the storage and retrieval of large volumes of unstructured, semi-structured or even structured data. The class of NoSQL DBMS' includes HBase that is the non-relational data store for Hadoop and MongoDB that was designed to support homogenous databases.
- R is an open source programming language and software environment that supports statistical computing and graphics. It has an increasing importance as a tool for computational statistics, visualization and data science.

3 Big Data Analytics

In this section we compare the methods and techniques of big data analytics. Data analytics means using formal mathematical techniques (e.g. statistical methods) to analyze large amounts of data and extract useful information. Many traditional data analytics methods are useful for big data analytics [19].
Table 2 shows a comparative Study of big data


analytics techniques.

4 Big Data Processing

In this section we compare some of the main big data processing methods and techniques [19].
Table 3 shows a comparative study of big data processing methods and techniques.

5 High Performance Computing

HPC means to practice aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business [1].
We use the HPC system of Bibliotheca Alexandrina which has a SUN cluster of peak performance of 11.8 Tflops, 130 eight-core compute nodes, 2 quad-core sockets per node, each is Intel Quad Xeon E5440 @ 2.83GHz, 8 GB memory per node, total memory 1.05 TBytes, 36 TB shared scratch, Node-node interconnect, Ethernet & 4x SDR Infiniband network for MPI, 4x SDR Infiniband network for I/O to the global Lustre filesystems.
A high speed, low latency clustering interconnect is essential for high-performance and scalability. The InfiniBand interconnect outperformed gigabit Ethernet by up to 115%. Furthermore, InfiniBand showed better scalability than gigabit Ethernet. With InfiniBand, WRF performance improved when additional compute nodes were added, while gigabit Ethernet showed little performance gain after 20 nodes. Faster WRF run times translate into improved performance/watt, optimizing power/performance criteria for power-aware simulations [14].
During a National Strategic Computing Initiative in United States of America the president Barack Obama orders some Policies and rules to maximize benefits of high-performance computing (HPC) research, development, and deployment [7].

6 Weather Research and Forecasting Model (WRF)

The Weather Research and Forecasting (WRF) model is a numerical weather prediction (NWP) and atmospheric simulation system designed for both research and operational applications [22].
The development of a WRF model had been a multi-agency effort to build a mesoscale forecast model and data assimilation system to advance the understanding and prediction of mesoscale weather and accelerate the transfer of research advances into operations. The initial and lateral boundary conditions used in the WRF pre-processing system (WPS) package was provided by the National Centers for Environmental Prediction (NCEP). Global final analyses on 1 X 1 degree resolution from Global Forecast System (GFS) analyses and lateral boundary conditions were updated every six hours, such as temperature, geopotential, wind, humidity, snow depth and cover. The WRF is suitable for a broad spectrum of applications across scales ranging from meters to thousands of kilometers. The Advanced Weather Research and Forecasting (WRF-ARW) model consists of three components, which can be found in the flow chart figure 1 [8]:

- WRF pre-processing System (WPS).
- WRF model.
- WRF Post-processing System.

The following are other NWP models which are available through NOAA's National Operational Model Archive and Distribution System (NOMADS) [6]:

- **Global Data Assimilation System (GDAS)** is the set of assimilation data, both input and output, in various formats for the Global Forecast System model. GDAS has been archived since 2004.
- **Global Ensemble Forecast System (GEFS)** is a global weather forecast model made up of 21 separate forecasts, or ensemble members, used to quantify the amount of uncertainty in a forecast. GEFS is produced four times a day with weather forecasts going out to 16 days.
- **Global Forecast System (GFS)** is a weather forecast model, composed of four separate models that work together to provide an accurate picture of weather conditions. The entire globe is covered by the GFS down to a horizontal resolution of 28 km.
- **North American Mesoscale (NAM)** is a regional weather forecast model covering North America down to a horizontal resolution of 12 km. Dozens of weather parameters are available from the NAM grids, from temperature and precipitation to lightning and turbulent kinetic energy.
- **Rapid Refresh (RAP)** is a regional weather forecast model of North America with separate sub-grids (with different horizontal resolutions) within the overall North America domain. RAP

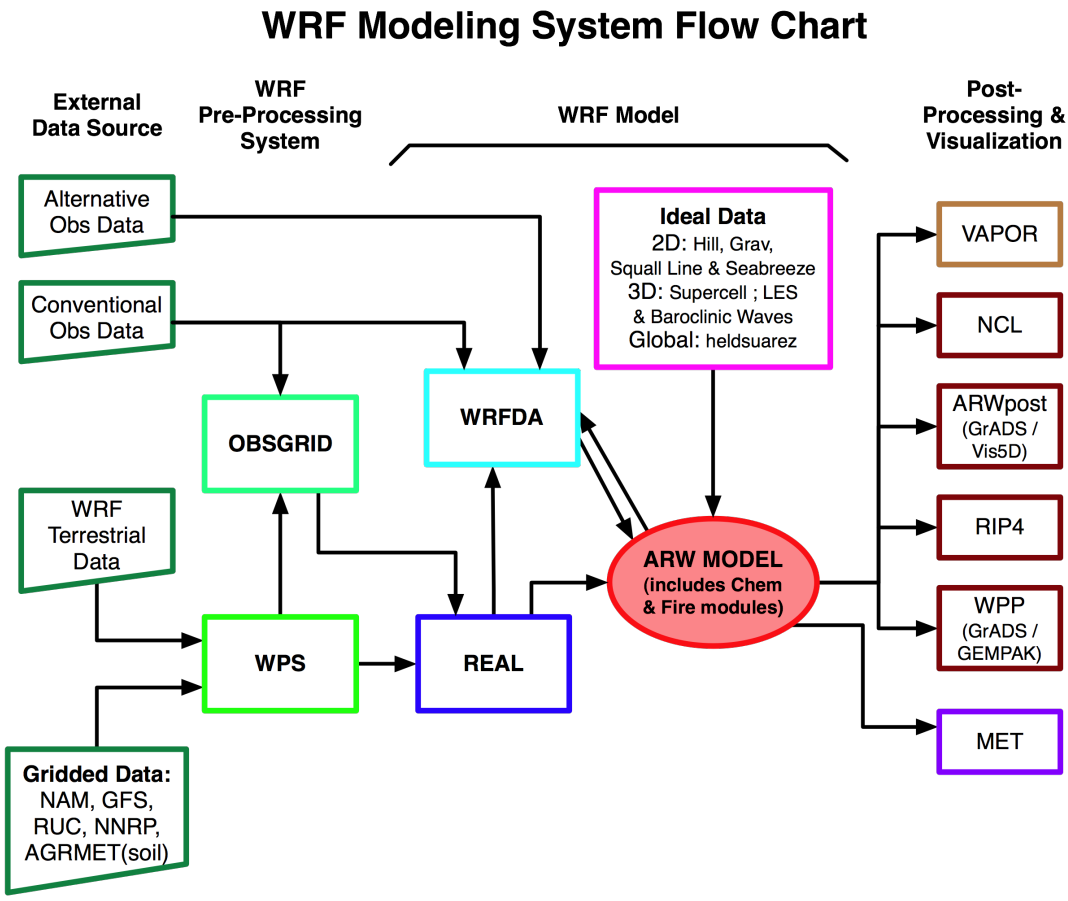



Figure 1: The WRF Modeling System

forecasts are generated every hour with forecast lengths going out 18 hours.

- **Rapid Update Cycle (RUC)** is a regional weather forecast model of the continental United States with forecast lengths going out 12 hours. RUC data are no longer produced operationally by the National Centers for Environmental Prediction (NCEP).

7 Related Works

Buszta and Mazurkiewicz (2015) built a new approach to weather forecasting that enhanced weather forecasting by data visualization and used neural networks to predict Climate Changes and analysis data[9].
Ibrahim and et al. (2014) used WRF Model to predict short range rainfall over Egypt and compared the actual rain gauge measurements at all stations over Egypt with WRF prediction and recommend using WRF in heavy rainfall prediction [23].
El Afandi and et al. (2013) studied Heavy Rainfall Simulation over Sinai Peninsula Using the Weather Research and Forecasting Model and approved that WRF model was able to predict rainfall in a significant consistency with real measurements [13].
El-Sammany (2010) forecast for Flash Floods over Wadi Watier Sinai Peninsula Using the Weather Research (WRF) Model and creates a comparison between WRF model results and real rainfall measurements [12].

POWERS (2007) used for the first time Weather Research and Forecasting (WRF) Model to Predict an Antarctic Severe Wind Event [21].

Mearns and et al. (1995) analyzed daily variability of precipitation in a nested regional Climate model and compare it to observations and doubled CO2 results [18].

Huang and et al.(1995) used a model to calculate and analyzed soil moisture over the united States (1931 - 1993) for a long-range temperature forecasting [15].


8 Weather Data Forecasting and Analysis Model

We build a new model to handle big data management phases, using big data technologies and tools. The new model helps to extract useful information from a large amount of data. Information helps data scientists to make decisions or suggestions.
Figure 2 shows the model outlines.

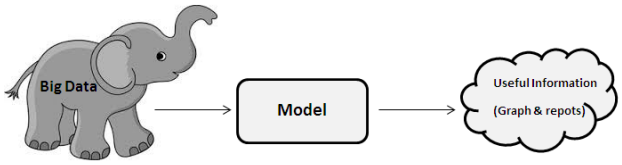


Figure 2: Model outlines

New model phases are shown in figure 3 and explain afterwards.

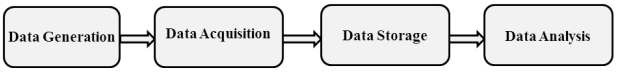


Figure 3: Model phases

- Phase 1: Data Generation, in this phase we need to generate a large amount of data such as weather data forecasting which is used in our case study. This phase can be used for any type of data like IoT data, Medical data etc. There are some requirements to generate weather data forecasting.
  - A weather forecasting simulation model like WRF. It requires high-performance computing systems. Commodity clusters have become very important for high performance computing due to the price for performance, flexibility and scalability they can deliver.
  - Availability of HPC (High Performance Computing).
- Phase 2: Data Acquisition, this phase includes data collection, data transmission, and data pre-processing. We assume that this phase is using the same hardware environment used in the data generation phase. Therefore, we consider data pre-processing in this phase only. We convert the output of WRF from the semi-structured NetCDF format to structured data format to facilitates management, storage and analysis.
- Phase 3: Data Storage, in this phase we use two Hadoop components to store and manage data; namely
  - HDFS (Hadoop Distribution File System) to store massive data for weather forecasting.
  - MapReduce to distribute processing on HPC clusters.
- Phase 4: Data Analysis, in this phase different statistical methods for analyzing and presenting the weather data and simulation results have been used. Histogram, cumulative distribution function (CDF), Time series are examples of these statistical methods.
  - Histogram is a useful tool to plot the data density. Histogram displays the tabulated frequency graphically as bars.
  - The cumulative distribution function (CDF) is the probability that the variable takes a value less than or equal to x. That is [2]

$$F(x) = Pr[X \leq x] = \alpha \tag{1}$$

For a continuous distribution, this can be expressed mathematically

$$F(x) = \int_{-\infty}^{x} f(t)dt \tag{2}$$

For a discrete distribution, the CDF can be expressed as

$$F(x) = \sum_{i=0}^{x} f(x) \tag{3}$$

  - A time series is a series of values, usually collected at regular time intervals. Time series data occur naturally in many application areas such as economics, financial, environmental$\cdots$ [10]. It is important to explore relationships among other variables (Temperature, Humidity$\cdots$).

We use an analysis programming language that supports analysis methods and visualization graphs; R over Hadoop. Also we can use a query language like Hive or pig over Hadoop and/or Sqoop to transfer bulk data between Hadoop and structured data stores.
Figure 4 shows the model structure.

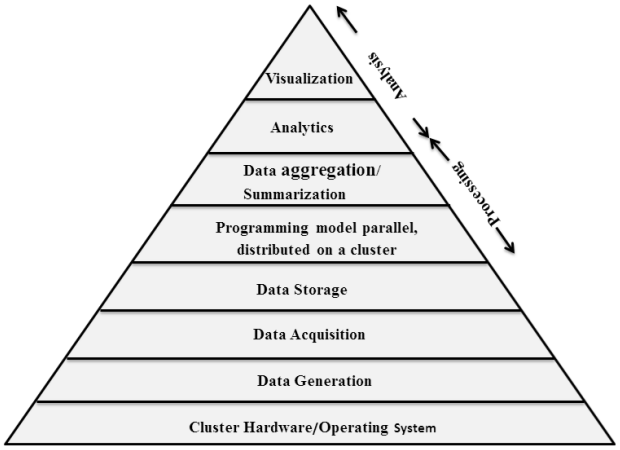



Figure 4: Model structure

Figure 5 shows the tools used in the model.

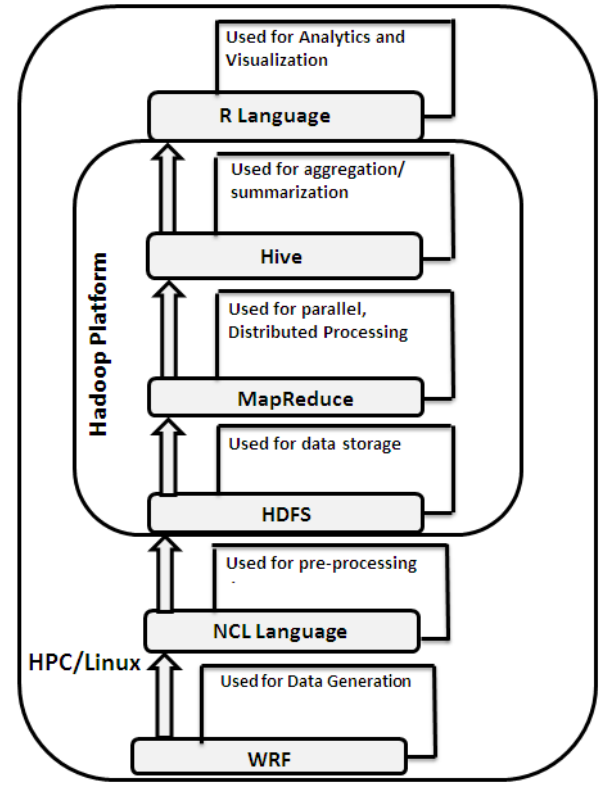


Figure 5: Tools used in the model

The table 4 explains the software packages used in the model and its versions.

Figure 6 shows the work flow in the model life cycle.

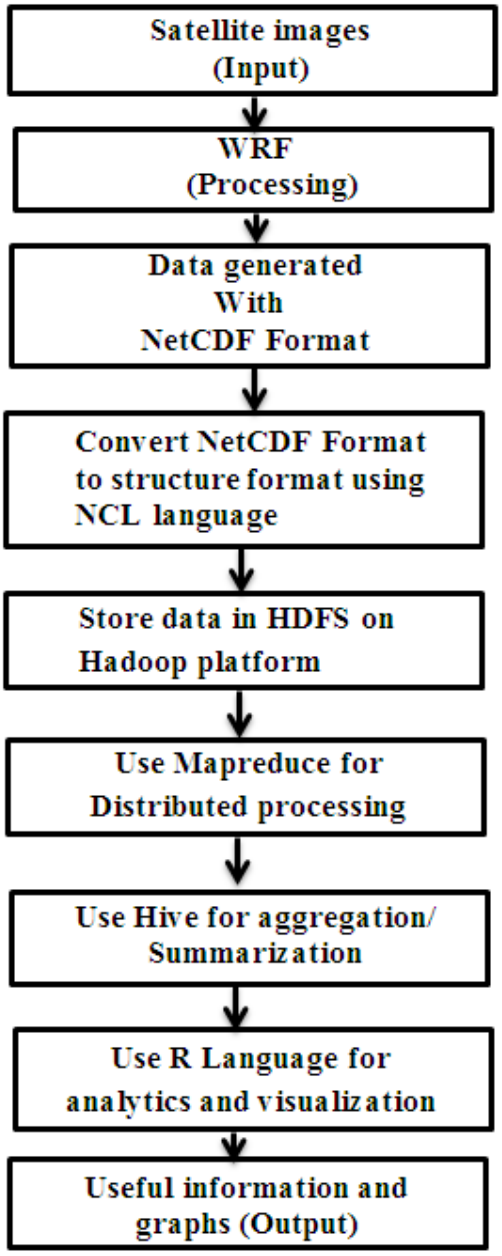


Figure 6: The model life cycle

9 Case of Study

The presented case of study generates weather data forecasting for a century. The above mentioned model is applied on climate data for a chosen area in Egypt. Multiple climate factors (temperature, wind speed, humidity, rainfall$\cdots$) are taken into consideration based on hourly calculations of WRF. To simulate one year using BA-HPC needs 15 clock days when 65 eight-core compute nodes are allocated. The total data set size generated for one year of simulation is 1.5 GB. The initial and lateral boundary meteorological data which was used to run the WRF model has


been downloaded from http://rda.ucar.edu/datasets. We used the above mentioned model to store, manage, and analyze data.

9.1 Results Analysis

Data summary shows that it is generated from 2000 to 2100.
Min. 1st Qu. Median Mean 3rd Qu. Max.
2000 2032 2054 2054 2077 2100
Figure 7 Shows a general view of a temperature data using Boxplot.

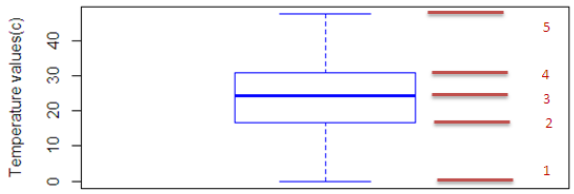


Figure 7: A box plot graph for temperature values

Figure 7 shows the following information about the data:

1. The minimum
2. The lower quartile (Q1)
3. The median (Q2)
4. The upper quartile (Q3)
5. The maximum

Figure 8 and figure 9 provide a general view of rainfall data with and without outlines.

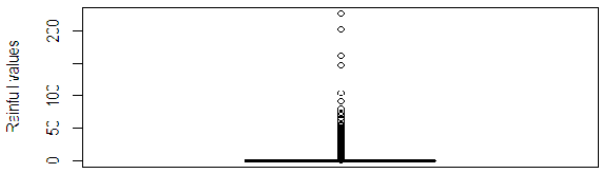


Figure 8: A box plot graph for rainfall values with outlines

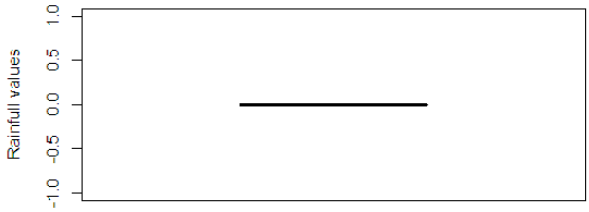


Figure 9: A box plot graph for rainfall values without outlines

Figure 9 shows that rain will not fall on this area for the coming century. However, the detailed figure 8 shows that the area will have few rains in the coming century.

Figure 10 and 11 get a general view of humidity data with and without outline.

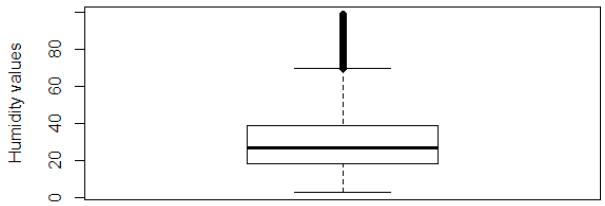


Figure 10: A box plot graph for humidity values with outlines

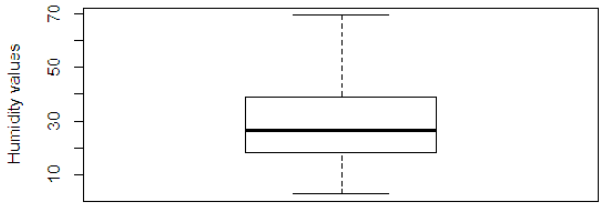


Figure 11: A box plot graph for humidity values without outlines

Figure 10 shows the area suffering from humidity higher than normal ranges. Analyzing the outline points shows that the humidity is proportional to the amount of rain.
Figures 12 , 13 and 14 show histograms for temperature, wind speed and relative humidity of climate data.

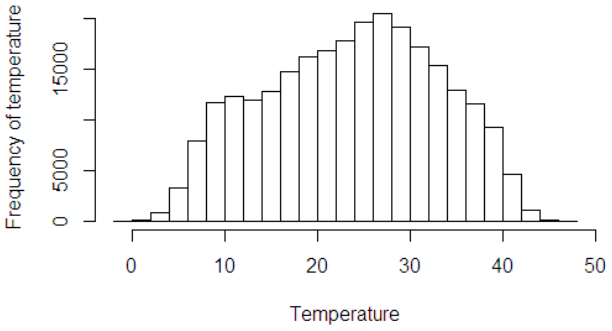


Figure 12: A histogram graph for temperature

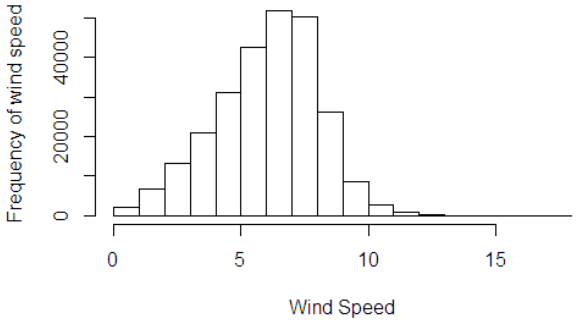


Figure 13: A histogram graph for wind speed

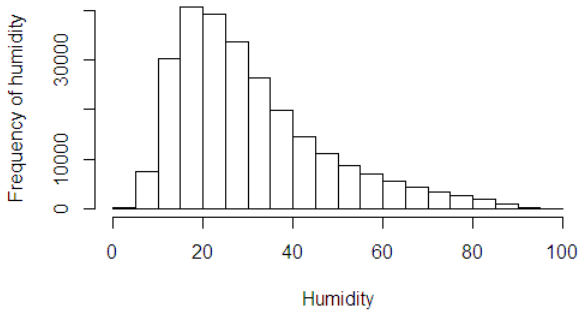


Figure 14: A histogram graph for humidity

Figures 15 and 16 show the cumulative distribution functions for temperature and relative humidity of climate.

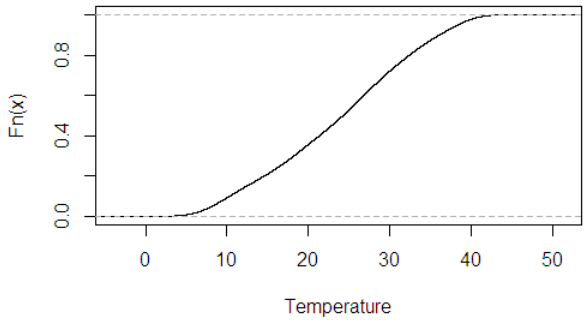


Figure 15: A cumulative distribution function graph for temperature

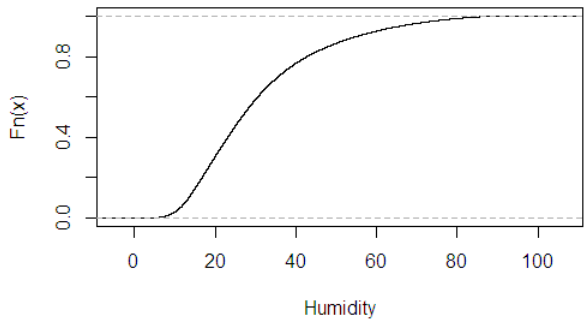


Figure 16: A cumulative distribution function graph for Humidity

Figures 17 and 18 show time series analysis for temperature and humidity.

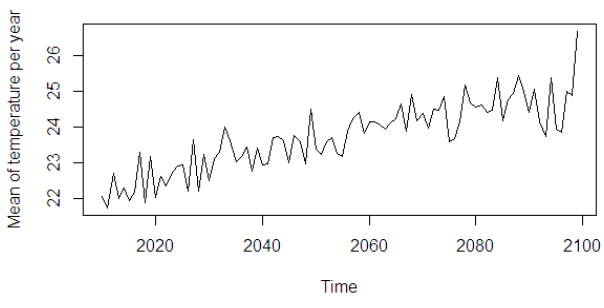


Figure 17: A time series graph for temperature

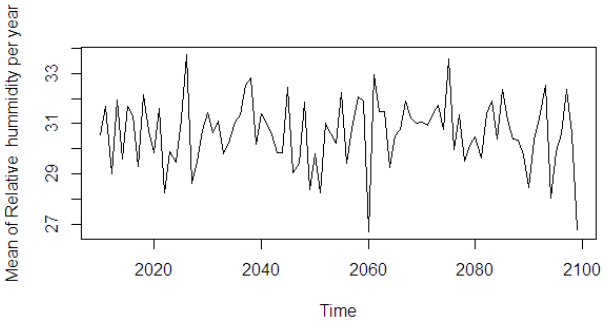


Figure 18: A time series graph for humidity

Figure 17 shows that the average of temperature will increase with 4 degrees during a century. While figure 18 shows that the humidity extreme points will be up and down in certain years.

10 Conclusions

This paper introduces big data concepts and focuses on two big data management phases; processing and analysis. We build a new model to study climate data changes. The model has been used to forecast, store and analyze climate data. From our analysis we observe that the average of temperature will increase with 4 degrees through a century. There are extreme points up and down for humidity in certain years, The humidity is proportional to the amount of the rain in outline points.

Future Work

The new model may be used to study other environmental changes including solar radiation, wind speed and earthquakes, calculating time of applying the model and comparing it with traditional ways. Anomalies of big data including noise, uncertainty and missing data (null values) may be handled.

References

[1] Inside HPC, 2011.

[2] Engeneering statistic handbook, Related Distributions, 2014.

[3] IBM-What is a Big Data, 2014.

[4] The Big Data Conundrum: How to Define It?,MIT Technology review, 2014.

[5] Top 500 supercomputers list, 2014.

[6] Numerical Weather Prediction, 2015.

[7] The white house, 2015.

[8] University corporation for atmospherich research, 2015.

[9] J. Mazurkiewicz A. Buszta. Climate Changes Prediction System Based on Weather Big Data Visualisation. *Advances in Intelligent Systems and Computing*, 365(1):75–86, 2015.

[10] P.J. Brockwell and R.A. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics, 1986.

[11] B. Devlin. Extracting Value from Chaosz, IDCs Digital Universe Study, 2011.

[12] M. S. El-Sammany. Forecasting of Flash Floods over Wadi Watier Sinai Peninsula Using the Weather Research and Forecasting WRF Model. *International Journal of Environmental, Chemical, Ecological, Geological and Geophysical Engineering*, 4(10):11, 2010.

[13] et al. G. El Afandi, M. Morsy. Heavy Rainfall Simulation over Sinai Peninsula Using the Weather Research and Forecasting Model. *International Journal of Atmospheric Sciences*, page 11, 2013.

[14] et al. G. Shainer, T. Liu1. Weather Research and Forecast (WRF) Model Performance and Profiling Analysis on Advanced Multi-core HPC Clusters. In *LCI Conference on high performance clustring computing*, 2009.

[15] et al. Huang. Analysis of Model Calculated Soil Moisture over the United States 1931Ű1993 and Applications to Long-Range Temperature Forecasts. *Journal of Climate*, 9(6):1350–1362, 1996.

[16] S. Tae Bae J. Kim, G. Wang. A Survey of Big Data Technologies and HowSemantic Computing Can Help. *International Journal of Semantic Computing*, 8(1):99–117, 2014.

[17] et al. J. Manyika, M. Chui. Big data: The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global, may 2011.

[18] et al. L.O. Mearns, F. Giorgi. Analysis of daily variability of precipitation in a nested regional Climate model comparison with observations and doubled CO2 results. *Global and Planetary Change*, 10(1):55–78, 1995.

[19] Y. Liu M. Chen, S. Mao. Big Data: A Survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[20] Dr.S.R.Gupta M.H.Padgavankar. Big Data Storage and Challenges. *International Journal of Computer Science and Information Technologies*, 5(2):2218–2223, 2014.

[21] J. G. POWERS. Numerical Prediction of an Antarctic Severe Wind Event with the Weather Research and Forecasting WRF Model. *Monthly Weather Review*, 135(9):3134–3157, 2007.

[22] et al. Q. Zhao. A snowmelt runoff forecasting model coupling WRF and DHSVM. *Hydrology Earth System Science*, 1(13):1897–1906, 2009.

[23] G. El Afandi S. Ibrahim. Short-range Rainfall Prediction over Egypt using the Weather Research and Forecasting Model. *open journal of renewable energy and sustainable development*, 1(2):56–70, 2014.

[24] D. Gupta S. Siddiqui. Big Data Process Analytics: A Survey. *International Journal of Emerging Research in Management Technology*, 3(7):117–123, 2014.

[25] K. Cukier V. Mayer-Schönberger. *Big Data: A Revolution that Will Transform how We Live, Work, and Think.* American Journal of Epidemiology, 2013.

Biographies

Prof. Kamal Abdelraouf ElDahshan He is a professor of Computer Science and Information Systems at Al-Azhar University in Cairo, Egypt. An Egyptian national and graduate of Cairo University, he obtained his doctoral degree from the Universitè de Technologie de Compiègne in France, where he also taught for several years. During his extended stay in France, he also worked at the prestigious Institute National de Tèlècommunications in Paris. Professor ElDahshanâĂŹs extensive international research, teaching, and consulting experiences have spanned four continents and include academic institutions as well as government and private organizations. He taught at Virginia Tech as a visiting professor; he was a Consultant to the Egyptian Cabinet Information and Decision Support Centre (IDSC); and he was a senior advisor to the Ministry of Education and Deputy Director of the National Technology Development Centre. Professor ElDahshan is a professional Fellow on Open Educational Resources as recognized by the United States Department of State and an Expert at ALECSO as recognized by the League of Arab States.

Hend Mancy She received her B.Sc. from Faculty of Science, AL-Azhar University at 2006.where she is working as a Demonstrator and Ms. Mancy is preparing M.Sc in the big data analysis under the supervision of prof.Kamal Eldahshan Her research interests include Big data, data science and data uncertainty.

Table 1: A comparative study of big data types

	Structured	**Semi-structured**	**unstructured**
Data size	From 5% to10% of data available around us.	From 5% to10% of data available around us.	80% of data available around us.
Example	Include RDBMS, data warehousing, and spreadsheets.	Include XML, HTML-tagged text.	Include e-mail messages, documents, videos, photos, audio files, logs.
Data storage	Data generated and stored in row, column format.	Data generated and stored in tags or other markers to capture elements.	Data generated without delimitation, punctuation or metadata. And cannot be stored in row, column format.

Table 2: A comparative Study of big data analytical techniques

	Type	**Definition**	**Features**
Cluster Analysis	Statistical method	Grouping objects, divide objects into several categories (clusters) according to their features.	Unsupervised technique
Factor Analysis	Statistical method	Grouping objects into factors. Few factors are used to extract the most useful information.	Data reduction
Correlation Analysis	Analytical method	Determining the law of relations, such as correlation, correlative dependence, and mutual restriction, among observed objects. Correlation coefficient is a measure of linear association between two variables and belongs to [-1,1].	
Regression Analysis	Mathematical method	Finding the relationships among one or more variables. Based on a group of experiments or observed data. It may make complex and undetermined relationships among variables to be easy.	Prediction
A/B Testing analysis	Statistical method	Called bucket testing .A method for determining how to improve target variables by comparing the tested group.	Big data will require a large number of tests to be executed and analyzed.
Statistical Analysis	Is based on the statistical theory, a branch of applied mathematics.	Can summarize and describe datasets or draw conclusions from data subject to random variations	Used in multiple fields such as environmental field.
Data Mining	Classification, clustering, regression, statistical learning, association analysis, and linking mining.	Is a process for extracting hidden, unknown useful information and knowledge from massive, incomplete, noisy, fuzzy, and random data.	Data mining algorithms C4.5, k-means, SVM, AprioriEM, Naive Bayes, and Cart, etc.


Table 3: A comparative study of big data processing methods and techniques

	Usage	**Advantage**	**Disadvantage**
Bloom Filter	Used to store hash values of data other than data itself by utilizing a bit array by using hash functions.Used for compression data storage.	Advantages in high space efficiency and high query speed.	Disadvantages in misrecognition and deletion.
Hashing	Used to transforms data into shorter fixed-length numerical values or index values.	Advantages in rapid reading, writing, and high query speed.	Hard to find a sound hash function.
Index	Used to reduce the expense of disk reading and writing. Used for all types of data (structured, semi-structured, unstructured).	Advantages in improving insertion, deletion, modification and query speeds.	Disadvantage in the additional cost of storing index files which should be maintained dynamically when data is updated.
Triel/ Trie Tree	Used to rapid retrieval and word frequency statistics. The main idea of Triel is to utilize common prefixes of character strings.	Advantages in reducing comparison on character strings to the greatest extent, so as to improve query efficiency.	
Parallel Computing	Used several computing resources to complete a computation task. Used to decompose a problem and assign them to several separate processes to be independently completed, so as to achieve co-processing. Parallel computing models include MPI (Message Passing Interface), MapReduce, and Dryad and you can see a comparison of this models in table 1 [19].	Advantage in reducing calculation time of task. Faster than sequential computing.	More hardware required.

Table 4: The software packages used in the model and its versions

Name	**Version**
The Advanced Research WRF (ARW) modeling system	Version 3.0
NCAR Command Language (NCL)	Version 6.3.0
Apache Hadoop Include (HFDS, MapReducce)	Version 2.6.0
Hive	Version 1.1.0
R language	Version 3.0.0